Guida avanzata a Solaris

Sun Microsystems Italia S.p.A.
Centro Direzionale Colleoni
Palazzo Andromeda, 1
20041 Agrate Brianza (MI)

N. parte: 801-7483-10
Revisione A, agosto 1994

Indice

Introduzione

A chi è destinato il manuale

Questo manuale si rivolge agli utenti del software di sistema Solaris™, costituito da SunOS™ e da OpenWindows™.

Prima di leggere il manuale

Prima di affrontare la lettura del manuale è necessario installare il sistema e predisporlo per l'uso. Per informazioni sulle procedure da seguire, vedere i manuali *SPARC: Installing Solaris Software* o *x86: Installing Solaris Software* o il manuale di installazione del proprio sistema.

Altro materiale di riferimento

L'AnswerBook del software di sistema Solaris consente di accedere a una serie di testi riguardanti Solaris, raggruppati nel modo seguente:

- Documentazione per amministratori di sistema (Sun Administrator's Set)
  Questo gruppo di manuali offre informazioni dettagliate sull'installazione e l'amministrazione del sistema con riferimento a numerose configurazioni, tra cui le reti di workstation Sun.

- Documentazione per sviluppatori (Sun Developer's Set)
  Questo gruppo di manuali è destinato agli sviluppatori di software; contiene informazioni relative alla scrittura, al debugging e alla manutenzione dei programmi utilizzati sui sistemi.

- Documentazione di riferimento (Sun Reference Manual Set)
  Questi manuali contengono una descrizione dei singoli comandi di SunOS. Le pagine di riferimento, chiamate anche pagine `man`, possono essere installate come documentazione online.

- Documentazione per l'utente
  Questi manuali offrono una descrizione dettagliata di vari aspetti del software di sistema SunOS, tra cui l'uso dei comandi di SunOS, l'uso di OpenWindows, la personalizzazione dell'ambiente di lavoro, la risoluzione dei problemi, la creazione di script, l'uso della posta elettronica e il lavoro in rete.

Per maggiori informazioni sull'Answerbook di Solaris, vedere il manuale *Solaris Roadmap*.

Convenzioni tipografiche

In questo manuale vengono adottate le seguenti convenzioni tipografiche:

- Il *corsivo* viene utilizzato per gli argomenti dei comandi, le variabili e i titoli dei manuali; viene inoltre utilizzato per fornire un'enfasi particolare a determinati termini all'interno del testo. Ad esempio:

  Inserire l'argomento *nomefile* come descritto in *man Pages(1): User Commands*.

- Il **`Courier neretto`** indica i comandi che devono essere digitati dall'utente durante una sessione di lavoro. Ad esempio:

```
$ date
```

- Il `Courier` viene usato per i listati dei programmi, i nomi dei comandi, i nomi dei programmi e i nomi dei sistemi. Ad esempio:

  Per diventare superutente, digitare `root` al prompt di login o digitare il comando `su` al prompt del sistema.

Viene inoltre utilizzato per il testo che il sistema visualizza sullo schermo. Ad esempio:

```
$ who
logos      console     set 11 15:36
```

- I bottoni del mouse vengono identificati in base alla funzione, non alla posizione, e sono indicati interamente in maiuscolo. Ad esempio si troverà indicato “Premere SELEZIONA”, e non “Premere il bottone sinistro”.
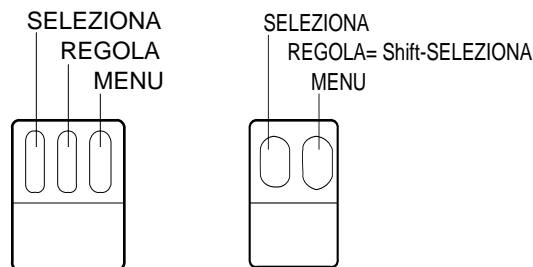
Figura P-1 Funzioni di default dei bottoni del mouse

Nel mouse a tre bottoni, vengono assegnate per default le seguenti funzioni:

- SELEZIONA = bottone sinistro
- REGOLA = bottone centrale
- MENU = bottone destro

Nel mouse a due bottoni, le funzioni assegnate per default sono:

- SELEZIONA = bottone sinistro
- MENU = bottone destro

Per utilizzare la funzione REGOLA con i mouse a due bottoni, tenere premuto il tasto Shift e fare clic con SELEZIONA.

- I nomi dei tasti sono indicati con l’iniziale maiuscola (ad esempio, Premere il tasto Shift). Su alcune tastiere il tasto Return è indicato con Enter o Invio.

Nota – Solaris supporta sia sistemi SPARC che piattaforme x86. Poiché i due tipi di sistema sono dotati di tastiere diverse, alcuni degli equivalenti da tastiera disponibili per i comandi di menu di OpenWindows differiscono a seconda del sistema utilizzato.

Per la maggior parte degli equivalenti da tastiera, Solaris utilizza una combinazione con il tasto Meta utilizzabile su entrambe le piattaforme. Un elenco dei comandi e dei relativi equivalenti da tastiera è disponibile nel Capitolo 1, “Login in SunOS e avvio di OpenWindows”.

- In questo manuale, i comandi di Open Windows eseguiti mediante equivalenti da tastiera (detti anche *acceleratori da tastiera*) vengono rappresentati utilizzando la convenzione seguente: un trattino (-) posto tra due o più tasti indica la pressione simultanea di quei tasti.

  Ad esempio, Ctrl-Alt significa che è necessario premere contemporaneamente i tasti Control e Alt; Ctrl-Alt-c significa che occorre premere i tasti Control e Alt e simultaneamente digitare una c.

- Gli equivalenti da tastiera si ottengono premendo il tasto Meta insieme a uno o più caratteri. Sulle tastiere SPARC, Meta corrisponde al tasto <>, mentre sulle tastiere x86 corrisponde alla pressione simultanea dei tasti Ctrl e Alt (Ctrl-Alt). Per maggiori informazioni sugli equivalenti da tastiera, vedere il Capitolo 1, "Login in SunOS e avvio di OpenWindows".
- La selezione di un'opzione di menu viene indicata nella forma seguente: selezionare Area di lavoro ➤ Programmi.
- Molti menu contengono un'opzione di default, indicata da un contorno ovale detto *anello di default*. Questa opzione viene selezionata automaticamente facendo clic con SELEZIONA sul bottone del relativo menu.

Convenzioni specifiche

I simboli seguenti, posti prima del titolo di una sezione o di una procedura, indicano che quanto descritto riguarda solo la piattaforma specificata:

SPARC   Indica una procedura o una sezione relativa a piattaforme SPARC

x86   Indica una procedura o una sezione relativa a piattaforme x86

Le note relative alle specifiche piattaforme vengono indicate come segue:

SPARC – Per note relative alle piattaforme SPARC.

---

x86 – Per note relative alle piattaforme x86.

I richiami alle piattaforme vengono indicati come segue:

- **SPARC:** Per richiami relativi alle piattaforme SPARC.
- **x86:** Per richiami relativi alle piattaforme x86.

Se un codice, una tabella o una figura si riferiscono a una piattaforma specifica, all'inizio del titolo viene indicata la piattaforma in oggetto. Ad esempio:

Tabella 1-1 SPARC: Titolo relativo a piattaforme SPARC

Tabella 1-2 x86: Titolo relativo a piattaforme x86

Login in SunOS e avvio di OpenWindows

1

Il software di OpenWindows deve essere installato sul disco rigido o su un server accessibile dal proprio sistema. Se non si è certi della collocazione di OpenWindows, rivolgersi all'amministratore del sistema o consultare i manuali *SPARC: Installing Solaris Software* o *x86: Installing Solaris Software*

In questo capitolo viene spiegato come effettuare il login nel sistema, come utilizzare un interprete di comandi, come attivare e avviare l'ambiente OpenWindows, come uscire dal sistema a finestre e come effettuare il logout. Vengono inoltre considerati alcuni casi particolari, come la configurazione dell'ambiente OpenWindows per due monitor.

Login

Una *sessione di lavoro standard* si definisce come l'intervallo di tempo che trascorre tra il login nel sistema e il logout. L'ambiente multiutente SunOS richiede che l'utente si identifichi ogni volta che accede al sistema. Il *nome di login* (detto anche *nome utente* o *profilo*) serve al sistema come chiave di identificazione dell'utente e per altre funzioni. La *password* limita l'uso del profilo alle persone che conoscono la chiave di accesso. Se ancora non si possiedono un nome di login e una password, chiedere all'*amministratore del sistema* di predisporli. Una volta in possesso di queste informazioni, si è pronti per effettuare il login.

Prima di effettuare il login nel sistema, verificare che sullo schermo compaia il seguente prompt:

```
login:
```

Inserire il nome di login ricevuto dall'amministratore del sistema e premere il tasto Return. Ad esempio, se il proprio nome di login è `stella`, digitare:

```
login: stella
```

e premere Return. Il sistema chiederà quindi la password di accesso:

```
login: stella
Password:
```

Inserire la propria password al prompt e premere Return. (Se al proprio profilo non è assegnata una password, il sistema consente l'accesso senza chiederla.) Si osservi che il sistema non visualizza la password sullo schermo mentre la si inserisce. Si tratta di un meccanismo di sicurezza per impedire che altri utenti possano scoprire la vostra password.

La shell di login

I capitoli seguenti contengono le istruzioni preliminari per l'utilizzo dei comandi di SunOS. Quando si inserisce un comando, in realtà si forniscono informazioni a un programma di interpretazione dei comandi, detto *shell*. La shell legge le informazioni ricevute e fa in modo che il sistema esegua l'operazione richiesta.

La shell di default di SunOS è la Bourne shell, ma nel sistema operativo sono disponibili altri due interpreti di comandi: la C shell e la Korn shell. Ognuno di questi programmi è dotato di proprie caratteristiche.

Nota – Per avere informazioni specifiche su un qualsiasi comando di SunOS, compresi i vari tipi di shell disponibili, è possibile consultare la pagina `man` relativa al comando di interesse. Per maggiori informazioni sulle pagine `man`, vedere "Visualizzazione delle pagine di riferimento con man" al Capitolo 2, "Comandi principali di SunOS".

Quando si effettua il login nel sistema (o si apre una nuova Finestra di comando o una finestra Shell), sullo schermo compare un prompt di sistema; la comparsa del prompt indica che è stato avviata automaticamente una shell, chiamata *shell di login*. Se la propria shell di login non è quella di default di SunOS (Bourne shell), significa che il sistema è stato configurato per l'utilizzo di una shell diversa (C shell o Korn shell).

Come già accennato, ogni shell è dotata di proprie caratteristiche. Alcuni comandi o procedure disponibili in una shell potrebbero non essere disponibili in un'altra. Se non specificato diversamente, i comandi e le procedure descritti in questo manuale si riferiscono di norma alla shell di default di SunOS (Bourne shell).

Logout

Per uscire dal sistema operativo al termine della sessione di lavoro, digitare il seguente comando di logout:

```
$ exit
```

Dopo qualche istante, il sistema tornerà a visualizzare il prompt di login:

```
$ exit
login:
```

La comparsa del prompt di login segnala che il logout è stato effettuato con successo. A questo punto il sistema è pronto a ricevere un nuovo login.

Nota – Con il sistema operativo SunOS, lo spegnimento della workstation o del terminale *non* comporta necessariamente il logout dal sistema. Se non si esegue la corretta procedura di logout, è possibile che la connessione instaurata al login non venga interrotta.

Equivalenti da tastiera

Anche se per la maggior parte delle operazioni è richiesto l'uso dei menu di OpenWindows e del mouse, in alcuni casi è possibile accelerarle utilizzando specifiche sequenze di tasti, detti *acceleratori da tastiera*, e alcuni tasti speciali preconfigurati.

La tabella seguente elenca alcuni comandi e i relativi equivalenti da tastiera per sistemi SPARC e x86.

Nota – Sulle tastiere SPARC, il componente Meta equivale al tasto <>, mentre sulle tastiere dei sistemi x86 si ottiene premendo Ctrl-Alt.

Per eseguire le operazioni con gli acceleratori da tastiera, tenere premuto il primo tasto (Meta o Control-Alt assieme) e digitare il secondo carattere richiesto. Ad esempio, per tagliare un testo selezionato, tenere premuto il tasto Meta e premere *x* su un sistema SPARC; su un sistema x86, tenere premuti i tasti Control e Alt e premere *x* simultaneamente.

Tabella 1-1 Acceleratori da tastiera

Operazione	Equivalente da tastiera	Azione
Ripeti	Meta - a	Ripete l'operazione precedente
Copia	Meta - c	Copia la selezione nella clipboard
Taglia	Meta - x	Taglia la selezione e la posiziona nella clipboard
Ricerca	Meta - f	Trova la selezione a destra del punto di inserimento
Guida	Help o F1	Visualizza una finestra di guida sensibile al contesto per l'elemento su cui è posizionato il puntatore

Tabella 1-1 Acceleratori da tastiera

Operazione	Equivalente da tastiera	Azione
Nuovo	Meta - n	Carica un nuovo file
Apri (file)	Meta - a	Apre un file (ad esempio, se si è evidenziata l'icona di un file in File Manager)
Apri/Chiudi (finestra)	Meta - w	Apre un'icona o riduce a icona una finestra
Incolla	Meta - v	Copia il contenuto della clipboard nel punto di inserimento
Stampa	Meta - p	Invia un file selezionato alla stampante (ad esempio, se si è evidenziata l'icona di un file in File Manager)
Caratteristiche	Meta - i	Visualizza la finestra Caratteristiche per l'applicazione su cui è posizionato il puntatore
Annulla cancellazione	Shift-Meta - p	Annulla un'operazione di annullamento
Salva	Meta - s	Salva il file corrente
Stop	Stop o Esc	Interrompe l'operazione corrente
Annulla	Meta - u	Annulla l'operazione precedente

Considerazioni sull'avvio di OpenWindows

Prima di avviare OpenWindows, il materiale di riferimento elencato qui di seguito può fornire informazioni utili per particolari configurazioni. Se le condizioni specificate non riguardano il proprio sistema, si passi al paragrafo "Visualizzazione di OpenWindows" in questo capitolo.

- Se occorre installare un server di OpenWindows in una rete, si consultino i manuali *SPARC: Installing Solaris Software* o *x86: Installing Solaris Software.*
- Se sul proprio sistema è attualmente installato OpenWindows versione 3, si veda l'Appendice A, "Migrazione a OpenWindows 3.3 o a versioni successive".

- Se sul proprio sistema è attualmente installato l'ambiente SunView, vedere l'Appendice A, "Migrazione a OpenWindows 3.3 o a versioni successive".
- Se sul proprio sistema è attualmente installata una versione di OpenWindows precedente alla 3.3, vedere i paragrafi "La variabile d'ambiente OPENWINHOME" e "Uso del file di avvio corretto" in questo capitolo.

x86 – In caso di sostituzione del mouse, della tastiera o della scheda video del sistema successivamente all'installazione di Solaris, sarà necessario eseguire il programma `devconfig` per aggiornare la configurazione del sistema. Per ulteriori informazioni, vedere la pagina man relativa a `devconfig`(1M).

La variabile d'ambiente `OPENWINHOME`

Se si utilizza una versione di OpenWindows precedente alla 3.3, è probabile che la configurazione corrente preveda l'uso della variabile d'ambiente `OPENWINHOME`. Con la nuova versione di OpenWindows questa variabile non deve più essere impostata, né manualmente né all'interno di un file di avvio.

Il comando `openwin` imposta automaticamente la variabile d'ambiente `OPENWINHOME` su `/usr/openwin`; questo rende superflua l'impostazione da parte dell'utente.

Se la variabile d'ambiente `OPENWINHOME` è stata impostata nel file `.profile` o `.cshrc` nella directory home, commentare o cancellare la riga che la contiene *prima* di avviare OpenWindows 3.3 o una versione successiva.

Per eliminare o commentare la variabile d'ambiente `OPENWINHOME` nel file `.profile` o `.cshrc`:

1. **Aprire il file `.profile` o `.cshrc` con un editor di testo, ad esempio `vi`.**
2. **Digitare un cancelletto (#) prima della variabile, come illustrato qui di seguito, o cancellare l'intera riga.**
   Se la variabile è impostata nel file `.profile`, eseguire la procedura `a`; se è impostata nel file `.cshrc`, eseguire la procedura `b`.

   a. **Nel file `.profile`:**

```
#OPENWINHOME=/usr/openwin
```

b. Nel file `.cshrc`**:**

```
#setenv OPENWINHOME /usr/openwin
```

3. **Salvare il file e uscire dall'editor.**

4. **Annullare l'impostazione della variabile d'ambiente** `OPENWINHOME` **(oppure effettuare un logout e quindi un login).**
   Se si utilizza una Bourne shell o una Korn shell, eseguire la procedura `a`. Se si utilizza una C shell, eseguire la procedura `b`.

   a. In una Bourne shell o in una Korn shell, digitare:

```
$ unset OPENWINHOME
```

   b. In una C shell, digitare:

```
esempio% unsetenv OPENWINHOME
```

Una volta annullata l'impostazione della variabile d'ambiente si è pronti per avviare OpenWindows. Vedere le relative istruzioni al paragrafo "Visualizzazione di OpenWindows" in questo capitolo.

Uso del file di avvio corretto

Se attualmente si utilizza una versione di OpenWindows anteriore alla 3.3, è importante determinare lo stato del file `.xinitrc`, collocato nella directory home. Il file `.xinitrc` è un file di avvio di OpenWindows che può contenere opzioni definite dall'utente.

Per determinare lo stato del file `.xinitrc`, digitare i comandi seguenti:

```
$ cd
$ ls -a .xinitrc
```

A seconda dell'output di questo comando, procedere in uno dei modi seguenti:

- Se non è presente un file `.xinitrc` (se cioè l'elenco ottenuto con il comando `ls -a` non contiene il nome del file), non fare nulla. Se la propria directory home non contiene il file `.xinitrc`, OpenWindows utilizza il file di avvio di default del sistema.
- Se il file `.xinitrc` è presente (se cioè l'elenco ottenuto con il comando `ls -a` contiene il nome del file) ma non è mai stato modificato, o comunque non si desidera conservare le modifiche apportate, procedere come descritto al punto 1 del paragrafo "Operazioni sui file di avvio".
- Se il file `.xinitrc` è presente (se cioè l'elenco ottenuto con il comando `ls -a` contiene il nome del file), è stato modificato e si desidera conservare le modifiche apportate, procedere come descritto al punto 2 del paragrafo "Operazioni sui file di avvio".

Operazioni sui file di avvio

1. **Per cancellare il file** `.xinitrc` **dalla propria directory home, digitare il comando seguente:**

```
$ rm .xinitrc
```

2. **Per conservare le modifiche apportate al file** `.xinitrc`**, procedere come segue:**

   a. **Rinominare** `.xinitrc` **in** `.xinitrc.save`:

```
$ mv .xinitrc .xinitrc.save
```

   b. **Copiare** `$OPENWINHOME/lib/Xinitrc` **in** `.xinitrc` **nella propria directory home:**

```
$ cp $OPENWINHOME/lib/Xinitrc $HOME/.xinitrc
```

   c. **Aggiungere le righe di** `.xinitrc.save` **che si desidera conservare in** `.xinitrc`**.**

Attenzione – Durante la modifica del file `.xinitrc`, non aggiungere una versione secondaria di `olwm`, non aggiungere `svenv` e non cancellare la riga che contiene `$OPENWINHOME/lib/openwin-sys`.

Avvio dell'ambiente OpenWindows

Per avviare l'ambiente OpenWindows, procedere come segue:

1. Se si utilizza NFS, attivare OpenWindows dal server sul quale è installato usando il comando `mount`.

   Per informazioni su come attivare OpenWindows da un server, vedere il manuale *OpenWindows Reference Manual*, oppure rivolgersi all'amministratore del sistema.

2. Avviare OpenWindows con il comando `openwin`, aggiungendo eventualmente le opzioni di avvio desiderate.

Visualizzazione di OpenWindows

Dopo avere attivato OpenWindows da un server e avere eseguito lo script di installazione, si è finalmente pronti per avviare OpenWindows.

Per avviare OpenWindows, digitare `/usr/openwin/bin/openwin` al prompt della shell e premere Return.

```
$ /usr/openwin/bin/openwin
```

Questo comando visualizza la videata iniziale di OpenWindows versione 3.3 e installa l'ambiente di lavoro.

Visualizzazione rapida di OpenWindows

Dopo avere verificato che OpenWindows si avvia correttamente, è possibile configurare il sistema in modo da utilizzare un comando di avvio più breve, per evitare di ripetere ogni volta l'intero percorso di OpenWindows.

Se si utilizza una Bourne shell o una Korn shell, è possibile inserire una funzione shell nel file `.profile`. Se si utilizza una C shell è possibile inserire un *alias* nel file `.cshrc`. Sia il file `.profile` che il file `.cshrc` si trovano nella directory home.

Una volta inserito il comando abbreviato nel file appropriato per la propria shell, per avviare OpenWindows sarà sufficiente digitare:

```
$ openwin
```

Per informazioni su come inserire il comando abbreviato nei file di avvio, vedere i paragrafi seguenti "File .profile" e "File .cshrc".

File `.profile`

Per inserire il comando abbreviato per l'avvio di OpenWindows nel file `.profile`:

1. **Aprire il file `.profile` usando un editor di testo, ad esempio `vi`.**
2. **Inserire la funzione shell seguente, scrivendola esattamente nella forma indicata:**

```
openwin () {
        /usr/openwin/bin/openwin
}
```

3. **Salvare il file e uscire dall'editor.**
4. **Effettuare un logout e quindi un nuovo login per attivare il comando, oppure digitare:**

```
$ . .profile
```

A questo punto, per avviare OpenWindows sarà sufficiente digitare `openwin`.

File `.cshrc`

Per inserire il comando abbreviato per l'avvio di OpenWindows nel file `.cshrc`:

1. **Aprire il file `.cshrc` usando un editor di testo, ad esempio `vi`.**
2. **Inserire il seguente comando di alias, scrivendolo esattamente nella forma indicata:**

```
alias openwin /usr/openwin/bin/openwin
```

3. **Salvare il file e uscire dall'editor.**
4. **Effettuare un logout e quindi un nuovo login per attivare il comando, oppure digitare:**

```
esempio% source .cshrc
```

A questo punto, per avviare OpenWindows sarà sufficiente digitare `openwin`.

Se OpenWindows non si avvia

OpenWindows viene avviato dalla directory `/usr/openwin`, dove è installato per default. Molte applicazioni, ad esempio Calendario, non possono essere caricate se il software di OpenWindows non è correttamente installato in `/usr/openwin`.

Se OpenWindows non si avvia dopo avere digitato il comando `/usr/openwin/bin/openwin`, è possibile che il software di OpenWindows non sia installato, o che sia installato in una directory diversa. In questo caso, consultare il manuale *OpenWindows Reference Manual* oppure rivolgersi all'amministratore del sistema.

Uscita dall'ambiente OpenWindows

Quando si lavora nell'ambiente a finestre OpenWindows, non è possibile effettuare il logout come si è soliti fare per terminare una sessione standard di SunOS. Occorre prima uscire dall'ambiente a finestre e quindi effettuare il vero e proprio logout.

Digitando `logout` al prompt di una shell, compare il messaggio:

```
Non è una shell di login.
```

Per uscire dall'ambiente OpenWindows, procedere come segue:

1. **Posizionare il mouse in modo che la freccia (*puntatore*) si trovi sullo sfondo dello schermo (l'*area di lavoro*).**
2. **Premere il bottone del mouse `MENU`.**
   Compare il menu Area di lavoro con le opzioni disponibili.
3. **Trascinare il puntatore lungo il menu fino a evidenziare l'ultima opzione, `Esci`.**
4. **Rilasciare il bottone del mouse.**
   Compare una finestra pop-up che chiede conferma dell'uscita dal sistema a finestre.
5. **Posizionare il puntatore su `Esci` e fare clic con il bottone del mouse `SELEZIONA`.**
   Dopo qualche istante, tutte le finestre verranno chiuse e nell'angolo inferiore sinistro dello schermo comparirà il prompt di sistema.

Opzioni speciali per l'avvio di OpenWindows

OpenWindows viene in genere avviato con la procedura descritta al paragrafo "Visualizzazione di OpenWindows". Tuttavia, in alcuni casi può essere opportuno utilizzare opzioni aggiuntive per modificare la procedura di avvio standard.

In questo paragrafo vengono descritti i seguenti casi particolari:

- Avvio di OpenWindows con sicurezza di rete ridotta.
- Avvio di OpenWindows con diversi tipi di monitor e frame buffer.
- Avvio di OpenWindows su più schermi.

Per avviare OpenWindows con una o più opzioni speciali, digitare il comando `openwin` come segue:

```
$ openwin [ opzioni ]
```

Nell'esempio precedente, *opzioni* indica le opzioni da inserire nella riga di comando per personalizzare la configurazione di default del server. Nei paragrafi seguenti sono descritte alcune delle opzioni di uso più comune.

Avvio con sicurezza di rete ridotta

Se si sta lavorando in un ambiente di rete aperto *e la sicurezza della rete non costituisce un problema*, è possibile utilizzare l'opzione `-noauth`, che consente ad altri utenti di eseguire le applicazioni residenti sul proprio sistema.

Il comando seguente prevale sulla funzione di sicurezza di default, che consente di specificare gli utenti con diritto di accesso al server:

```
$ openwin -noauth
```

Avvio con diversi tipi di monitor

Se si dispone di un monitor a scala di grigi (un monitor monocromatico con un frame buffer di 8 o più bit) è possibile avviare OpenWindows con l'opzione `grayvis`. Questa opzione non è necessaria, ma permette di migliorare alcuni aspetti della visualizzazione di OpenWindows.

Per utilizzarla, digitare quanto segue al prompt di sistema:

```
$ cd
$ openwin -dev /dev/fb grayvis
```

SPARC Avvio con più monitor

SPARC – Questo paragrafo riguarda soltanto i sistemi basati su processori SPARC. La procedura descritta richiede una certa esperienza di amministrazione dei sistemi. Se non si è mai eseguita una procedura di configurazione, rivolgersi all'amministratore di sistema.

Per avviare OpenWindows su più schermi è necessario comunicare al sistema i nomi dei dispositivi aggiuntivi e i tipi di visualizzazione che si desidera utilizzare. È possibile specificare le opzioni relative ai dispositivi, oppure utilizzare i valori di default disponibili per lo script di avvio `openwin`.

Per avviare OpenWindows su due monitor occorre specificare due opzioni insieme al comando `openwin`:

```
$ openwin [ [ -dev dispositivo ] [ opzioni_dispositivo ] ]
```

Le doppie parentesi indicano che la combinazione di [`-dev` *dispositivo*] e [*opzioni_dispositivo*] può essere inserita più volte nella riga di comando (una volta per ogni dispositivo).

[-dev *dispositivo*]

L'opzione *dispositivo* specifica il frame buffer, o schermo, che il server dovrà utilizzare per la visualizzazione.

Se questa opzione non viene specificata, il server utilizza il dispositivo di default, `/dev/fb`. Se l'opzione `-dev` viene specificata più volte nella riga di comando, essa indica l'uso di più monitor con lo stesso server.

[*opzioni_dispositivo*]

Le *opzioni_dispositivo* sono una serie di modificatori in grado di cambiare il comportamento del dispositivo specificato con l'opzione `-dev`.

Esempi di opzioni per i dispositivi

Questo paragrafo contiene alcuni esempi in cui vengono utilizzati due monitor, in posizione affiancata o l'uno sopra l'altro.

Nota – In tutti gli esempi è importante notare l'ordine dei dispositivi. Il primo dispositivo specificato deve essere lo schermo disposto fisicamente a sinistra o sopra il secondo. Il secondo dispositivo deve essere lo schermo collocato fisicamente a destra o sotto il primo.

[left] [right]

La seguente riga di comando ordina al sistema di avviare OpenWindows su due schermi. Quello di sinistra è il frame buffer di default, mentre quello di destra è uno schermo monocromatico. Questa configurazione permette di spostare il cursore a sinistra o a destra tra uno schermo e l'altro.

```
$ openwin -dev /dev/fb left -dev /dev/fbs/bwtwo0 right
```

L'esempio che segue è equivalente a quello precedente. Per default, il primo dispositivo specificato nella riga di comando si trova a sinistra del secondo.

```
$ openwin -dev /dev/fb -dev /dev/fbs/bwtwo0
```

La riga di comando seguente ordina al sistema di avviare OpenWindows su due schermi. Quello di destra è il frame buffer di default, mentre quello di sinistra è uno schermo monocromatico. Questa configurazione permette di spostare il cursore a sinistra e a destra tra uno schermo e l'altro.

```
$ openwin -dev /dev/fb right -dev /dev/fbs/bwtwo0 left
```

[top] [bottom]

La riga di comando seguente ordina al sistema di avviare OpenWindows su due schermi. Quello superiore è un CG6, mentre quello inferiore è un monitor monocromatico. Questa configurazione permette di spostare il cursore in alto e in basso tra uno schermo e l'altro.

```
$ openwin -dev /dev/fbs/cgsix0 top -dev /dev/fbs/bwtwo0 bottom
```

L'esempio che segue *non* è equivalente a quello precedente. Per default, il primo dispositivo specificato nella riga di comando si trova a *sinistra* del secondo.

```
$ openwin -dev /dev/fbs/cgsix0 -dev /dev/fbs/bwtwo0
```

La riga di comando seguente ordina al sistema di avviare OpenWindows su due schermi. Quello inferiore è un CG6, mentre quello superiore è un monitor monocromatico. Questa configurazione permette di spostare il cursore in alto e in basso tra uno schermo e l'altro.

```
$ openwin -dev /dev/fbs/cgsix0 bottom -dev /dev/fbs/bwtwo0 top
```

Alcune note aggiuntive

Se si utilizzano più schermi, è opportuno ricordare quanto segue.

- Per default, `olwm` gestisce tutti gli schermi.
- Non è possibile spostare le finestre tra uno schermo e l'altro.

Comandi principali di SunOS 2

Questo capitolo costituisce un'introduzione ai comandi principali del sistema operativo SunOS. Vengono descritte le procedure da seguire per digitare i comandi, correggere gli errori di battitura, eseguire comandi lunghi o multipli con le relative opzioni, ed altre informazioni sui comandi di SunOS.

Per eseguire i comandi, usare una Finestra di comando o una Shell, disponibili dal sottomenu Programmi dal menu Area di lavoro.

Il prompt di sistema

Dopo avere effettuato il login, lo schermo o la finestra contengono soltanto un prompt iniziale. Il tipo di prompt dipende dal tipo di shell in uso e dalla configurazione impostata dall'amministratore del sistema. Negli esempi di questo manuale verrà utilizzato il prompt di sistema di default di SunOS, il simbolo del dollaro ($).

Per informazioni su come cambiare il prompt di sistema, vedere "Modifica del prompt di sistema" al Capitolo 10, "Personalizzazione dell'ambiente di lavoro".

2

Esecuzione dei comandi

Quando sullo schermo compare il prompt della shell, significa che il sistema è pronto per ricevere un comando dall'utente. Provare a digitare il comando `date`, come indicato nell'esempio (digitare `date` e premere il tasto Return):

```
$ date
lunedì, 3 febbraio 1993, 10:12:51 MET
$
```

Come si può osservare, questo comando visualizza la data e l'ora corrente. Si provi ora ad inserire lo stesso comando con l'iniziale maiuscola:

```
$ Date
Date: Comando non trovato.
$
```

Come si può osservare, la `D` maiuscola non viene interpretata dal sistema allo stesso modo della `d` minuscola. Quasi tutti i comandi del sistema operativo SunOS devono essere digitati usando lettere minuscole.

Correzione degli errori di battitura

Se si è iniziato a digitare un comando, ad esempio `Date`, e ci si accorge di avere commesso un errore, è possibile correggerlo prima di premere il tasto Return. Il comando inserito, infatti, viene inviato al sistema solo premendo il tasto Return e prima di allora non ha alcun effetto. Per correggere un errore di battitura si può procedere in due modi:

- Premere il tasto Canc o Back Space per cancellare il testo fino al punto da correggere; oppure
- Digitare Ctrl-U per cancellare l'intera riga e ripetere l'intero comando. (Premere il tasto Control e contemporaneamente il tasto "u".)

Fare una prova con entrambi i metodi per valutarne gli effetti. (Il tasto Canc/Back Space può variare a seconda dei sistemi. La combinazione Ctrl-U dovrebbe funzionare su quasi tutti i sistemi.)

Esecuzione di comandi multipli o lunghi

Il sistema operativo SunOS permette di eseguire più comandi raggruppandoli in un'unica riga di comando. È sufficiente separare i comandi con un punto e virgola (;), come indicato nell'esempio con i comandi `date` e `logname`:

```
$ date; logname
lunedì, 3 febbraio 1993, 10:12:51 MET
stella
$
```

Come si può osservare, il sistema visualizza la data e l'ora corrente (risultato del comando `date`) e il nome di login dell'utente correntemente connesso al sistema (risultato del comando `logname`).

Se il comando da digitare è molto lungo, è possibile utilizzare una barra rovesciata (\) per continuare a scrivere sulla riga successiva. Ad esempio:

```
$ date; \
logname
lunedì, 3 febbraio 1993, 10:12:51 MET
stella
$
```

I comandi `date` e `logname` utilizzati nell'esempio, pur non essendo lunghi, dimostrano in modo semplice la possibilità di inserire una serie di comandi su più righe. Questa caratteristica risulta assai utile con un uso più avanzato del sistema, quando i comandi da inserire sono spesso più lunghi della larghezza dello schermo.

Nota – Se si utilizza una Shell o una Finestra di comando nell'ambiente OpenWindows, per digitare i comandi su più righe non c'è bisogno di ricorrere alla barra rovesciata. Quando si raggiunge la fine di una riga, i comandi inseriti passano automaticamente alla riga successiva; l'intera sequenza viene quindi eseguita premendo il tasto Return.

Ripetizione di comandi precedenti

Nota – La funzione di ripetizione dei comandi descritta in questo paragrafo è disponibile solo se si utilizza una C shell.

Un metodo rapido per ripetere l'ultimo comando eseguito consiste nel digitare `!!` e premere Return. Il sistema tiene in memoria una *storia* dei comandi eseguiti in precedenza ed è in grado di ripeterli. Ad esempio, se l'ultimo comando eseguito è `date`, digitando `!!` si otterrà quanto segue:

```
esempio% !!
date
lunedì, 3 febbraio 1993, 10:12:51 MET
esempio%
```

È anche possibile ripetere un qualsiasi comando precedente digitando `!`*x*, dove *x* è il numero corrispondente al comando desiderato nella *storia* registrata dal sistema. Per visualizzare la lista dei comandi memorizzati, digitare il comando `history` e premere Return. L'esempio seguente mostra un possibile output di questo comando:

```
esempio% history
1  pwd
2  clear
3  ls -l
4  cd /usr/home/worker
5  logname
6  date
7  history
```

Un altro metodo per ripetere un comando memorizzato consiste nel digitare `!` seguito da un numero negativo. Ad esempio, per ripetere il penultimo comando elencato nella storia, digitare quanto segue:

```
esempio% !-2
logname
stella
esempio%
```

Con riferimento alla storia elencata in precedenza, il sistema ha ripetuto il comando `logname`.

Un ulteriore metodo consiste nel digitare ! seguito dal primo carattere di un comando precedente. Ad esempio, se era stato inserito il comando `clear` per cancellare il contenuto dello schermo, lo stesso comando può essere ripetuto digitando `!cl`. Con questo metodo di ripetizione, tuttavia, occorre inserire un numero sufficiente di caratteri per rendere il comando identificabile in modo univoco. Se si inserisce una sola lettera dopo !, il sistema ripete il comando più recente che inizia con quella lettera.

Aggiunta di opzioni ai comandi

Molti comandi dispongono di *opzioni* che corrispondono a funzioni speciali. Ad esempio, il comando `date` dispone dell'opzione `-u`, che esprime l'ora nel fuso di Greenwich (GMT, Greenwich Mean Time) anziché nel fuso orario locale:

```
$ date -u
Lun, feb 3 10:12:51 GMT 1993
$
```

La maggior parte delle opzioni vengono espresse con un singolo carattere preceduto da un trattino (`-`). Non tutti i comandi dispongono di opzioni, mentre altri ne possiedono diverse. Per utilizzare più opzioni di uno stesso comando, è possibile digitarle sia separatamente (`-a -b`) che insieme (`-ab`).

Redirezionamento e piping dell'output di un comando

Se non specificato diversamente, i risultati dei comandi vengono visualizzati sullo schermo. Alcuni simboli speciali permettono tuttavia di *redirigere* l'output di un comando. Ad esempio, è possibile salvare l'output in un file anziché visualizzarlo sullo schermo. L'esempio seguente illustra l'uso del simbolo di redirezionamento (>):

```
$ date > file_di_prova
$
```

In questo esempio, l'output del comando `date` viene rediretto in un nuovo file di nome `file_di_prova`. Successivamente, il contenuto di `file_di_prova` può essere visualizzato con il comando `more`:

```
$ more file_di_prova
lunedì, 3 febbraio 1993, 10:12:51 MET
$
```

Come si può osservare, `file_di_prova` contiene l'output del comando `date`. (Per informazioni sul comando `more`, vedere il Capitolo 3, "File e directory".)

In alcuni casi è possibile redirigere l'output di un comando perché venga utilizzato come input da un altro comando. Una serie di comandi collegati in questo modo viene detta *pipeline*. Il simbolo utilizzato per questo tipo di redirezionamento è una barra verticale (`|`) detta *pipe*.

Ad esempio, anziché salvare l'output di un comando in un file, è possibile utilizzarlo come input per un comando di stampa (`lp`) usando il simbolo di pipe (`|`). Per inviare l'output del comando `date` direttamente alla stampante, digitare quanto segue:

```
$ date | lp
$
```

L'output del comando `date` verrà stampato. (Per informazioni sull'uso del comando di stampa `lp`, vedere "Invio di una richiesta di stampa alla stampante di default" al Capitolo 8, "Uso delle stampanti".)

Gli esempi di piping e redirezionamento illustrati finora sono piuttosto semplici, ma le potenzialilità di queste funzioni verranno ulteriormente approfondite nella descrizione di procedure più avanzate.

Esecuzione di comandi in background

In molti casi può essere comodo digitare un comando dal prompt di sistema e quindi lasciare che l'esecuzione avvenga in *background*. Se un comando non viene posto in background, il prompt non ricompare fino al termine della sua esecuzione. Perciò, per evitare i possibili tempi di attesa, si può rendere la finestra immediatamente disponibile per l'inserimento di altri comandi.

Per eseguire un processo in background, digitare una ‘e commerciale’ (&) dopo il comando, come illustrato nell’esempio seguente. Il numero che compare è l’identificativo (id) del processo:

```
$ bigjob &
[1] 21414
$
```

Il comando `bigjob` verrà eseguito in background, lasciando la shell immediatamente disponibile per l’inserimento di altri comandi. Una volta completato il processo, inserendo un nuovo comando, ad esempio `date`, verrà visualizzato un messaggio simile al seguente:

```
$ date
lunedì, 3 febbraio 1993, 10:12:51 MET
[1]  + Done    bigjob
$
```

Se si desidera terminare la sessione di lavoro senza attendere il completamento di un processo in background, usare il comando `nohup` (no hangup) per consentirne l’esecuzione completa, come illustrato in questo esempio. In caso contrario, il processo in background verrà terminato al momento del logout:

```
$ nohup bigjob &
[1] 21414
$
```

Come ottenere informazioni sui comandi OS

In questo paragrafo vengono descritte alcune funzioni di guida online. Queste funzioni permettono di ottenere informazioni di riferimento direttamente dalla propria workstation o dal proprio terminale.

Nota – Le funzioni descritte in questo paragrafo *si aggiungono* alle funzioni di guida di OpenWindows.

Visualizzazione delle pagine di riferimento con `man`

Se si conosce il nome di un comando ma non si è certi degli effetti che produce, il comando `man` può essere d'aiuto. Per avere maggiori informazioni su questo comando, digitare:

```
$ man man
```

Questo comando visualizza l'inizio di una pagina di riferimento del manuale online di SunOS. Premere la barra spaziatrice per passare alla videata successiva, oppure premere il tasto Q per uscire e tornare al prompt di sistema. Il comando `man` permette di ottenere informazioni su tutte le opzioni disponibili e sulla corretta sintassi dei comandi. Spesso le pagine di riferimento contengono esempi che illustrano i possibili impieghi dei comandi.

Visualizzazione di una breve descrizione dei comandi con `whatis`

Per visualizzare una descrizione sintetica del funzionamento di un comando, usare il comando `whatis` come illustrato in questo esempio:

```
$ whatis date
date (1)          -display or set the date
$
```

Si osservi il numero tra parentesi dopo il nome del comando. Questo numero indica la sezione del manuale di riferimento online alla quale appartiene il comando. I comandi sono infatti raggruppati in diverse categorie a seconda della loro funzione. La maggior parte dei comandi utente è contenuta nella sezione 1. Per convenzione, il numero della sezione compare tra parentesi dopo il nome del comando. Nel manuale di riferimento stampato, i comandi compaiono in ordine alfabetico all'interno del gruppo di appartenenza.

Ricerca per parole chiave con `apropos`

Se si vuole eseguire una data operazione ma non si è certi sul comando da utilizzare, il comando `apropos` può essere di aiuto. Questo comando permette di individuare uno o più comandi tramite una ricerca per parole chiave. Il risultato del comando `apropos` è un elenco di tutti i comandi le cui descrizioni

sintetiche contengono la parola chiave specificata. Se la parola chiave impostata figura in numerose descrizioni, il risultato potrà essere un lungo elenco di comandi in qualche modo correlati.

Per avere alcuni esempi dell’output del comando apropos, provare ad eseguire uno o più dei comandi seguenti:

- apropos who
- apropos execute
- apropos apropos

Se la parola chiave inserita genera un elenco eccessivamente lungo, è possibile premere Ctrl-C per interrompere l’elencazione e tornare al prompt di sistema. (Premere il tasto Control e contemporaneamente il tasto “c”.)

File e directory 3

La riga di comando di SunOS viene usata per eseguire operazioni su file e directory in maniera “manuale”, digitandone il nome insieme al comando SunOS appropriato. Si tratta di un approccio diverso da quello dell’applicazione File Manager di OpenWindows, in cui i file vengono visualizzati in forma di icone e la selezione dei comandi avviene tramite menu.

Questo capitolo introduce i concetti principali e le procedure da seguire per operare su file e directory dalla riga di comando di SunOS. Le operazioni descritte possono essere effettuate da qualsiasi riga di comando di SunOS, si tratti di shell o finestre di comando dell’ambiente OpenWindows o di terminali remoti. Per utilizzare efficacemente il sistema operativo SunOS è essenziale comprendere i concetti presentati in questo capitolo.

Informazioni generali sui file

Il *file* è l’unità fondamentale del sistema operativo SunOS. Quasi tutti i tipi di dati vengono gestiti come file, tra cui:

- **Documenti** — File di testo, come lettere o relazioni, sorgenti di programmi o qualsiasi altro tipo di dati che sia possibile scrivere e salvare.
- **Comandi** — La maggior parte dei comandi sono file *eseguibili*, vale a dire file che è possibile eseguire per avviare un determinato programma. Un esempio di un file eseguibile è il comando `date` esaminato al capitolo precedente, in quanto esegue un programma che restituisce la data corrente.

- **Dispositivi** — Tutti i dispositivi come terminali, stampanti o unità disco vengono gestiti come file.
- **Directory** — Una directory non è altro che un file che contiene altri file.

Il paragrafo seguente descrive i comandi disponibili per creare, elencare, copiare, spostare e cancellare i file. Verrà inoltre spiegato come visualizzare il contenuto di un file e come determinarne la natura.

Comandi per la gestione dei file

Per ogni comando descritto in questo paragrafo è fornito un esempio che ne illustra l'utilizzo. Può essere utile provare ad eseguire gli esempi sulla propria workstation: l'esercitazione pratica sui comandi permetterà di comprenderli e ricordarli più facilmente.

Prima di iniziare

Prima di iniziare l'esercitazione sui file, portarsi nella propria directory *home*. Si tratta della directory che l'amministratore del sistema genera appositamente per l'utente quando crea il suo profilo. Eseguendo le operazioni descritte nella propria directory home sarà più difficile creare, copiare, spostare e soprattutto cancellare i file di altri utenti.

Per spostarsi nella propria directory home, digitare il comando `cd` senza altri argomenti. Questo comando ha come effetto lo spostamento nella directory home dell'utente che lo esegue. Digitare quindi il comando `pwd` per visualizzare la posizione corrente all'interno del file system, cioè la propria directory home:

```
$ cd
$ pwd
/export/home/nomeutente
```

In questo esempio, la directory home dell'utente è `/export/home/`*nomeutente*, in cui *nomeutente* è il nome dell'utente proprietario della directory home.

Creazione di un file di testo

Per creare un nuovo file si utilizza il comando `touch`. Se non esiste un file con il nome specificato, il comando `touch` ne crea uno vuoto. Se invece esiste già un file con lo stesso nome, `touch` aggiorna l'ora dell'ultimo accesso a quel file.

```
$ touch filetemp
$
```

Elencazione dei file (`ls`)

Eseguire ora il comando `ls` per verificare la creazione del file:

```
$ ls filetemp
filetemp
```

Inserendo il comando `ls` da solo, vengono elencati tutti i file contenuti nella directory corrente. Digitando `ls` insieme al nome di un file, viene mostrato solo il file che corrisponde al nome inserito.

Per maggiori informazioni sul comando `ls`, fare riferimento a *man Pages(1): User Commands*.

Copia dei file (`cp`)

Usare il comando `cp` per copiare `filetemp` in un file chiamato `filecopia`:

```
$ cp filetemp filecopia
$
```

Provare ora a visualizzare entrambi i file. Si noti che entrambi i nomi iniziano con la stringa “file”. Questa condizione permette di utilizzare il *carattere speciale* asterisco (*), un simbolo che può sostituire qualsiasi carattere o sequenza di caratteri. Digitando `ls file*`, il sistema visualizzerà sia `filetemp` che `filecopia` (e qualsiasi altro file della stessa directory che inizi con `file`):

```
$ ls file*
filecopia    filetemp
```

Il nome `filecopia` compare per primo perché i file vengono elencati in ordine alfabetico. (Le lettere maiuscole e i numeri precedono le lettere minuscole.)

Per maggiori informazioni sul comando `cp`, vedere *man Pages(1): User Commands*.

Spostamento e rinomina dei file (`mv`)

Il comando `mv` permette sia di spostare che di rinominare un file. In questo esempio, il comando `mv` viene utilizzato per rinominare `filetemp` in `filevuoto`:

```
$ mv tempfile filevuoto
$
```

Visualizzare nuovamente i due file per verificare la modifica:

```
$ ls file*
filecopia    filevuoto
```

Come si può osservare, `filetemp` è stato sostituito da `filevuoto`.

Per maggiori informazioni sul comando `mv`, vedere *man Pages(1): User Commands*.

Cancellazione dei file (`rm`)

A questo punto, utilizzare il comando `rm` per cancellare `filecopia`; verificare quindi il risultato dell'operazione con il comando `ls`:

```
$ rm filecopia
$ ls file*
filevuoto
```

Attenzione – Un file cancellato non può più essere recuperato, a meno che non ne esista una copia di backup. Fare perciò molta attenzione nell'uso del comando `rm`, soprattutto se associato al carattere speciale "*". I file cancellati con `rm` non possono essere in alcun modo recuperati.

Per maggiori informazioni sul comando `rm`, vedere *man Pages(1): User Commands*.

Visualizzazione del contenuto di un file (`more`, `cat`)

Il comando `more` permette di visualizzare il contenuto di un file. Digitare `more` seguito dal nome del file che si desidera visualizzare. Il contenuto del file scorrerà lungo lo schermo. Se il file è più lungo di una videata, compare il seguente messaggio:

```
--Ancora--(nn%)  [premere spazio per continuare, 'q' per uscire.]
```

dove *nn* è la percentuale del file visualizzata fino a quel punto.

Anche il comando `cat` permette di visualizzare il contenuto di un file, ma in questo caso l'intero contenuto scorre sullo schermo in un'unica passata, senza pause tra una videata e l'altra. Il comando `cat` viene usato in genere per unire due o più file in un unico file più grande, come nell'esempio seguente:

```
$ cat file1 file2 file3 > fileunico
$ ls file*
file1
file2
file3
fileunico
$
```

Per maggiori informazioni sui comandi `more` o `cat`, vedere *man Pages(1): User Commands*.

Identificazione del tipo di file (`file`)

Alcuni file, tra cui i file binari e gli eseguibili, non possono essere né stampati né visualizzati sullo schermo. Il comando `file` può essere d'aiuto per identificare la natura di un determinato file.

In questo esempio, il comando `file` viene utilizzato per determinare la natura di `filecopia`:

```
$ file filecopia
filecopia:    testo ascii
```

Directory e gerarchie

Finora è stato spiegato come elencare, copiare, rinominare e cancellare i file. A questo punto si possono esaminare questioni più ampie. Ad esempio, dove sono situati i file? In questo paragrafo viene descritta la struttura gerarchica delle directory. Si consiglia di leggere attentamente la parte descrittiva e quindi provare ad eseguire gli esempi illustrati nei paragrafi successivi.

Struttura gerarchica delle directory

I file sono raggruppati in directory, che a loro volta sono organizzate in una struttura gerarchica. Al livello più alto della gerarchia si trova la directory "radice" o "root", rappresentata da una barra "`/`".

Come illustrato nella Figura 3-1, ogni directory del file system può contenere a sua volta molte directory. Per convenzione, i diversi livelli delle directory vengono separati con una barra `/`. Si osservi ad esempio che la directory root (`/`) contiene, tra le altre, le sottodirectory `/usr`, `/bin`, `/home` e `/lib`. La sottodirectory `/home` contiene a sua volta `utente1`, `utente2` e `utente3`.

Per specificare una directory (o un file al suo interno) occorre indicare i nomi delle directory in cui essa è contenuta. Questo tipo di indicazione viene detta *percorso*. Ad esempio, il percorso della directory `utente3` che compare nella figura è `/home/utente3`.
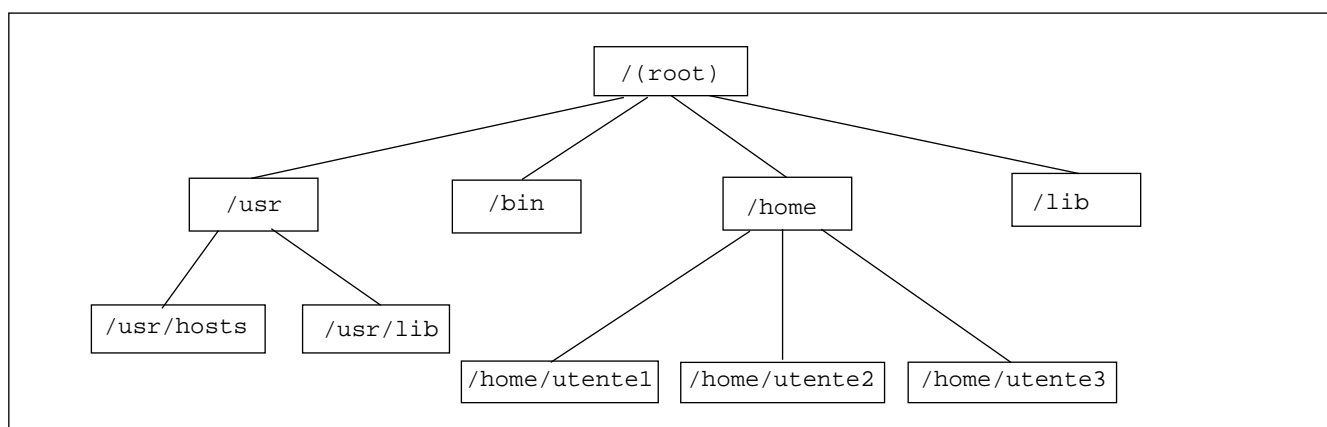
Figura 3-1 La struttura gerarchica del file system

Tutte le sottodirectory e i file contenuti in una stessa directory devono avere un nome unico, che li identifichi in modo esclusivo. È invece possibile che due o più file contenuti in directory diverse abbiano lo stesso nome. Ad esempio, la directory `/usr` contiene la sottodirectory `/usr/lib`. Tra `/usr/lib` e `/lib` non vi sono conflitti in quanto i percorsi sono diversi.

I percorsi dei file hanno esattamente la stessa struttura dei percorsi delle directory. Il percorso di un file descrive la sua collocazione nella gerarchia del file system. Ad esempio, se la directory `/home/utente2` contiene un file di nome `relazione5`, il percorso del file sarà `/home/utente2/relazione5`. Questo percorso indica che il file `relazione5` si trova nella directory `utente2`, la quale è contenuta nella directory `home`, la quale è contenuta a sua volta nella directory root (`/`).

Le directory possono contenere solo sottodirectory, solo file o entrambi.

Visualizzazione della directory corrente (`pwd`)

Il comando `pwd` visualizza la posizione attuale dell'utente nella gerarchia del file system:

```
$ pwd
/home/utente1
```

L'esempio precedente ha valore puramente indicativo, in quanto l'output del comando dipende dalla struttura delle directory del sistema. La posizione corrente nella gerarchia del file system viene detta directory corrente.

La directory home

Ogni utente dispone di una propria directory *home*. Aprendo una Finestra di comando o una Shell nell'ambiente OpenWindows, la posizione iniziale è sempre la directory home. Questa directory viene predisposta dall'amministratore del sistema al momento della creazione del profilo utente.

Spostamento in un'altra directory (`cd`)

Il comando `cd` permette di spostarsi in un'altra directory della gerarchia del file system:

```
$ cd /usr/lib
$ pwd
/usr/lib
```

Digitato senza argomenti, il comando `cd` riporta l'utente alla sua directory home. Se ad esempio la directory home è `/home/utente1`, il comando ha l'effetto seguente:

```
$ cd
$ pwd
/home/utente1
```

Nelle C shell, si può utilizzare il carattere tilde (~) come abbreviazione per indicare una directory home. Ad esempio, per spostarsi nella sottodirectory `musica` sotto la propria directory home si potrà digitare:

```
esempio% cd ~/musica
```

La tilde può essere utilizzata anche per specificare la directory home di un altro utente. Ad esempio, digitando:

```
esempio% cd ~nomeutente
```

dove *nomeutente* è il nome di login di un altro utente, ci si sposterà nella directory home di quell'utente.

Nota – Se si utilizza una Bourne shell, il carattere ~ non ha lo stesso effetto.

Se si utilizza la Bourne shell, è possibile che l'amministratore di sistema abbia configurato la workstation in modo che il comando `$home` specifichi la directory home dell'utente. In questo caso, digitando:

```
$ $home/musica
```

ci si sposterà nella sottodirectory `musica` della propria directory home. Analogamente, digitando:

```
$ $homenomeutente
```

dove *nomeutente* rappresenta il nome di login di un altro utente, ci si sposterà nella directory home dell'utente specificato.

La directory immediatamente "sopra" una sottodirectory viene detta *directory di livello superiore*. Nell'esempio precedente, `/home` è la directory di livello superiore rispetto a `/home/utente1`. Il livello superiore viene indicato con il simbolo `..` ("punto-punto"). Il comando `cd ..` produce perciò uno spostamento nella directory di livello superiore, come illustrato nell'esempio seguente:

```
$ pwd
/home/utente1
$ cd ..
$ pwd
/home
```

Se la directory corrente è `/home/utente1` e si desidera operare su un file contenuto in `/home/utente2`, è possibile utilizzare il comando seguente:

```
$ pwd
/home/utente1
$ cd ../utente2
$ pwd
/home/utente2
```

Il percorso `../utente2` ordina al sistema di cercare `utente2` nella directory di livello superiore. L'uso di questo comando abbreviato permette di non scrivere l'intero percorso `/home/utente2`.

Creazione di una directory (`mkdir`)

La creazione di una nuova directory è un'operazione molto semplice. È sufficiente digitare il comando `mkdir` seguito dal nome della nuova directory:

```
$ mkdir fiori
$ cd fiori
$ mkdir primule
$ cd primule
$ pwd
/home/utente2/fiori/primule
```

Percorso relativo

Il percorso completo di una directory o di un file inizia con una barra (`/`) e descrive l'intera struttura della directory tra il file o la directory specificata e la directory radice. In molti casi è tuttavia possibile utilizzare un percorso più breve che indichi la posizione *relativa* del file o della directory rispetto alla directory di lavoro corrente.

Per spostarsi in una sottodirectory della directory corrente è sufficiente indicare il nome della directory di destinazione senza specificarne il percorso. Nell'esempio precedente, il comando `cd fiori` utilizza il percorso relativo della directory di destinazione. Il suo percorso completo, supponendo che la directory corrente sia `/home/utente2`, sarebbe infatti `/home/utente2/fiori`.

Provare a creare diverse sottodirectory e a spostarsi dall'una all'altra. Usare sia i percorsi completi che i percorsi relativi e verificare di volta in volta la posizione corrente con il comando `pwd`.

Spostamento e rinomina delle directory

Per rinominare una directory, si esegue un comando che “sposta” la directory originale in un’altra directory di nome diverso. Il comando da utilizzare è `mv`:

```
$ pwd
/home/utente2/fiori
$ ls
primule
$ mv primule viole
$ ls
viole
```

Il comando `mv` permette anche di spostare una directory all’interno di un’altra directory:

```
$ pwd
/home/utente2/fiori
$ ls
viole
$ mv viole ../fiori2
$ ls ../fiori2
viole
```

In questo esempio, la directory `viole` viene spostata da `fiori` a `fiori2` utilizzando il comando `mv`.

Copia di una directory

Il comando `cp -r` permette di copiare una directory e i file che essa contiene:

```
$ cp -r fiori fiori3
$
```

Questo comando copia tutti i file e le sottodirectory contenute nella directory `fiori` in una nuova directory `fiori3`. Questa operazione viene detta copia *ricorsiva*, e viene specificata con l’opzione `-r`. Se si cerca di copiare una directory senza usare questa opzione viene visualizzato un messaggio di errore.

Cancellazione di una directory (`rmdir`)

Per cancellare una directory vuota, usare il comando `rmdir` come segue:

```
$ rmdir fiori3
$
```

Se la directory specificata contiene file o sottodirectory, il comando `rmdir` non la cancella.

Per cancellare una directory e tutto il suo contenuto, comprese eventuali sottodirectory e i rispettivi file, occorrerà utilizzare `rm -r` (aggiungendo cioè l'opzione *ricorsiva* `-r`). Digitare quanto segue:

```
$ rm -r fiori3
$
```

Attenzione – Le directory cancellate con il comando `rmdir` *non* possono essere recuperate, *né* possono essere recuperate le directory cancellate con il comando `rm -r`.

Confronto fra più file (`diff`)

Accade spesso che diverse persone che hanno accesso ad un file ne effettuino una copia e la modifichino. Il comando `diff` permette di identificare le differenze esistenti tra due versioni di un file ASCII. Il comando:

```
$ diff primo_file secondo_file
```

confronta riga per riga `primo_file` e `secondo_file` cercando eventuali differenze. Quando il programma individua una o più righe diverse, esso determina se la differenza riscontrata è il risultato di un'aggiunta, di una cancellazione o di una modifica, e individua il numero di righe interessate. L'output del comando indica i numeri delle righe che presentano differenze in ciascuno dei file, seguito dalle parti di testo che risultano diverse.

Se la differenza è il risultato di un'aggiunta, `diff` visualizza una riga nella forma seguente:

p[,*p*] a *s*[,*s*]

in cui *p* è il numero di una riga di `primo_file` e *s* è il numero di una riga di `secondo_file`.

Se la differenza è il risultato di una cancellazione, `diff` utilizza una `d` al posto della `a`; se è il risultato di una modifica, `diff` utilizza una `c`.

Le righe diverse dei due file seguono immediatamente i numeri di riga. Il testo di `primo_file` è preceduto da un simbolo di minore (<), mentre il testo di `secondo_file` è preceduto da un simbolo di maggiore (>).

Questo esempio mostra due file e l'output ottenuto con il comando `diff`:

```
$ cat agenda.7.15
Settimana del 7/15

Giorno:  Ora:        Azione:                Note:

M        10:00       Team hardware         ogni due settimane
M        1:30        Team software
G        3:00        Team documentazione
V        1:00        Colloquio
$ cat agenda.7.22
Settimana del 7/22

Giorno:  Ora:        Azione:                Note:

L        8:30        Riunione staff         tutto il giorno
M        10:00       Team hardware         ogni due settimane
M        1:30        Team software
G        3:00        Team documentazione
$ diff agenda.7.15 agenda.7.22
1c1
< Settimana del 7/15
---
> Settimana del 7/22
4a5
> L        8:30        Riunione staff         tutto il giorno
8d8
< V        1:00        Colloquio
```

Se i due file confrontati sono identici, il comando `diff` non genera alcun output.

Il comando `diff` dispone di molte altre opzioni oltre a quelle descritte. Per maggiori informazioni, vedere *man Pages(1): User Commands*.

Confronto fra tre file diversi (`diff3`)

Per individuare le differenze fra tre versioni di un file, usare il comando `diff3` come segue:

```
$ diff3 file1 file2 file3
```

`diff3` confronta tre versioni di un file e visualizza le parti di testo differenti marcandole con i seguenti codici:

==== tutti i file sono diversi

====1 *file1* è diverso

====2 *file2* è diverso

====3 *file3* è diverso

Uso di `bdiff` su file di grandi dimensioni

Se occorre confrontare file di grandi dimensioni, è consigliabile usare il comando `bdiff` anziché `diff`. La sintassi dei due comandi è la stessa:

```
$ bdiff primo_file secondo_file
```

Si consiglia di usare `bdiff` al posto di `diff` per file più lunghi di 3500 righe.

Ricerca di un file (`find`)

Il comando `find` ricerca i file che soddisfano le condizioni specificate dall'utente a partire dalla directory indicata. Ad esempio, è possibile cercare i file il cui nome contenga una determinata stringa o che siano stati modificati in un arco di tempo stabilito.

A differenza di molti comandi, le opzioni di `find` sono formate da più caratteri, e il nome della directory di partenza deve precedere le opzioni sulla riga di comando:

```
$ find directory opzioni
```

dove *directory* è il nome della directory di partenza e *opzioni* sono le opzioni del comando `find`.

Ogni opzione definisce un criterio per la selezione del file. Il programma seleziona solo i file che soddisfano tutti i criteri specificati. Aumentando il numero delle opzioni si restringe perciò il campo dei file corrispondenti. L'opzione `-print` indica che il risultato della ricerca deve essere visualizzato sullo schermo. (Come verrà descritto più avanti, il comando `find` può essere usato anche per l'esecuzione di comandi; in questo caso può essere superfluo visualizzare i file selezionati.)

L'opzione `-name` *nomefile* ordina a `find` di selezionare i file che corrispondono al *nomefile* specificato. In questo caso *nomefile* rappresenta l'ultimo componente del percorso del file. Ad esempio, l'ultimo componente del percorso `/usr/lib/calendario` è `calendario`. Questa parte del percorso di un file viene spesso definita *nome base*.

Ad esempio, per ricercare tutti i file della directory corrente e delle relative sottodirectory il cui nome termina in `s`, digitare quanto segue:

```
$ find . -name '*s' -print
./alias
./alias/files
./alias/files/gks
./src/gks
$
```

Qui di seguito sono descritte alcune delle opzioni più comuni:

`-name` *nomefile*

Seleziona i file il cui ultimo componente corrisponde a *nomefile*. Per indicare un modello di sostituzione anziché il nome del file, inserire *nomefile* fra virgolette singole.

`-user` *id_utente*

Seleziona i file di proprietà di *id_utente*. *id_utente* può essere un nome di login o il numero dell'ID utente.

`-group` *gruppo*

Seleziona i file di proprietà del *gruppo*.

`-m time` *n*

Seleziona i file che sono stati modificati negli ultimi *n* giorni.

`-newer` *file_di_riferimento*

Seleziona i file modificati più recentemente del *file_di_riferimento*.

È anche possibile specificare un ordine di precedenza, inserendo le opzioni tra parentesi precedute dal carattere di escape "`\`" (ad esempio, `\(`*opzioni*`\)`). All'interno delle parentesi è possibile inserire tra le opzioni il flag `-o`, per indicare a `find` di selezionare i file che corrispondono a ogni singolo criterio specificato anziché solo i file che soddisfano entrambi i criteri:

```
$ find . \( -name AAA -o -name BBB \) -print
./AAA
./BBB
```

È possibile invertire il senso di un'opzione facendola precedere da un punto esclamativo (a sua volta preceduto da un carattere di escape). In questo modo `find` seleziona i file che *non* corrispondono all'opzione specificata:

```
$ find . \!-name BBB -print
./AAA
```

È inoltre possibile utilizzare `find` per eseguire sui file selezionati una determinata operazione, usando l'opzione

`-exec` *comando* `'{}' \;`

Questa opzione termina con un punto e virgola preceduto dal carattere di escape (`\;`). Le parentesi graffe fra virgolette vengono sostituite dai nomi dei file selezionati da `find`.

Ad esempio, si può usare `find` per cancellare automaticamente i file temporanei. Se si utilizza un criterio coerente per denominare i file temporanei, si potrà usare `find` per cercarli e cancellarli ovunque si trovino. Ad esempio, se i file temporanei vengono chiamati `prova` o `esempio`, il comando seguente permetterà di ricercarli e cancellarli automaticamente:

```
$ find . \( -name prova -o -name esempio \) -exec rm '{}' \;
```

Per maggiori informazioni, vedere *man Pages(1): User Commands*.

Sicurezza di file e directory

Nota – Leggere questo paragrafo con attenzione. Una chiara comprensione delle autorizzazioni dei file è importante per molte operazioni quotidiane.

Le autorizzazioni dei file sono un metodo per proteggere file e directory dalla lettura e dalla scrittura da parte di utenti non autorizzati. In alcuni casi può essere opportuno autorizzare altri utenti a leggere determinati file ma non a modificarli. In altri casi si possono avere sul proprio sistema file eseguibili (programmi) da condividere. Le autorizzazioni consentono di controllare il tipo di accesso degli altri utenti ai propri file.

I tipi di autorizzazioni disponibili per file e directory sono i seguenti:

- `r` – autorizzazione di *lettura*. È necessaria per esaminare e copiare un file o per elencare il contenuto di una directory.
- `w` – autorizzazione di *scrittura*. È necessaria per modificare, cancellare o rinominare un file, o per aggiungere o eliminare un file da una directory.
- `x` – autorizzazione di *esecuzione*. È necessaria per utilizzare un file eseguibile, ad esempio un programma, o per accedere a una directory.

Le autorizzazioni possono essere impostare per tre categorie di utenti:

- Proprietario – l'utente
- Gruppo – altri utenti appartenenti allo stesso gruppo dell'utente (ad esempio tutti gli utenti di uno stesso reparto). I gruppi vengono creati dall'amministratore del sistema.
- Altri – qualsiasi altro utente

Visualizzazione delle autorizzazioni e dello stato (ls -l)

In precedenza è stato descritto l'uso del comando ls per l'elencazione dei file. Abbinando al comando ls l'opzione -l, è possibile ottenere un elenco dei file in formato *lungo*. I file e le directory vengono elencati in ordine alfabetico. La Figura 3-2 illustra questo metodo di visualizzazione:

```
$ pwd
/home/nomehost/utente2
$ ls -l
totale 8
drwxr-xr-x  2 utente2          1024 feb  9 14:22 directory1
-rw-r--r--  1 utente2             0 feb 10 10:20 filevuoto
-rw-r--r--  1 utente2        104357 feb  5 08:20 fileunico
drwxr-xr-x  3 utente2          1024 feb 10 11:13 fiori2
```

Autorizzazioni Collegamenti Proprietario Dimensione Data Ora File o directory

Figura 3-2 Visualizzazione delle autorizzazioni e dello stato di file e directory

Il primo carattere della riga indica il tipo di file. Un trattino (-) indica un file, una d una directory, mentre altri caratteri possono indicare altri tipi di file speciali.

I nove caratteri successivi indicano le autorizzazioni del file o della directory. La sequenza è formata da tre gruppi di tre caratteri ciascuno, indicanti rispettivamente le autorizzazioni del proprietario, del gruppo e degli altri utenti. Le autorizzazioni di filevuoto, rw-r--r--, indicano che il proprietario può leggere e scrivere il file, il gruppo e gli altri utenti lo possono leggere mentre nessuno lo può eseguire. Le autorizzazioni della directory fiori2, rwxr-xr-x, indicano che tutti gli utenti possiedono l'autorizzazione in lettura ed esecuzione sulla directory, mentre solo il proprietario ha l'autorizzazione in scrittura.

Oltre alle autorizzazioni dei file, l'elenco visualizzato contiene le seguenti informazioni:

- Numero dei collegamenti del file o della directory
- Nome del proprietario (utente2 in questo caso)
- Numero di byte (caratteri) del file
- Data e ora dell'ultimo aggiornamento del file o della directory
- Nome del file o della directory

Spostarsi nella propria directory home con il comando `cd` e provare ad eseguire il comando `ls -l`. (Il risultato sarà naturalmente diverso da quello illustrato nell'esempio.)

Provare ora a digitare il comando seguente:

```
$ ls -l nomedir
```

dove *nomedir* è il nome di una directory effettivamente presente nel file system. Indicando il nome di una directory, il comando `ls -l` visualizza le informazioni relative a tutti i file e directory contenuti nella directory specificata.

Elencazione dei file "nascosti" (`ls -a`)

Alcuni tipi di file non vengono elencati utilizzando il semplice comando `ls`. I nomi di questi file iniziano con il carattere `.` ("punto"), come ad esempio `.cshrc`, `.login` e `.profile`. Per elencare anche i "file punto", usare il comando `ls -a`:

```
$ ls -a
.
..
.cshrc
.login
.profile
filevuoto
```

Si noti che i file che iniziano con il carattere `.` sono elencati prima degli altri. L'elenco illustrato contiene due file speciali: il primo è il file `.`, che rappresenta il riferimento alla directory corrente, il secondo è il file `..`, che rappresenta il riferimento alla directory di livello superiore.

In genere, i file che iniziano con il carattere `.` vengono usati dai programmi di utilità del sistema e non vengono modificati dall'utente (a parte alcune eccezioni).

Modifica delle autorizzazioni (chmod)

Il comando `chmod` permette di cambiare le autorizzazioni di un file o di una directory. Questa operazione può essere eseguita solo dal proprietario del file o della directory o da un utente root. La sintassi generale del comando `chmod` è la seguente:

```
chmod  autorizzazioni  nome
```

dove *autorizzazioni* indica le autorizzazioni da modificare e *nome* è il nome del file o della directory su cui si desidera eseguire l'operazione.

Le autorizzazioni possono essere specificate in vari modi. Qui di seguito viene illustrata una delle forme di indicazione più semplici:

1. Usare una o più lettere per indicare gli utenti interessati:
   - `u` (*user* - il proprietario)
   - `g` (*group* - il gruppo)
   - `o` (*other* - altri utenti)
   - `a` (*all* - tutte le categorie)
2. Indicare se le autorizzazioni devono essere aggiunte (+) o eliminate (-).
3. Usare una o più lettere per indicare le autorizzazioni di interesse:
   - `r` (*read* - la lettura)
   - `w` (*write* - la scrittura)
   - `x` (*execute* - l'esecuzione)

Nell'esempio seguente, viene aggiunta l'autorizzazione di scrittura alla directory `viole` per gli utenti appartenenti allo stesso gruppo del proprietario (*autorizzazioni* diventa perciò `g+w` e *nome* viene sostituito con `viole`):

```
$ ls -l viole
drwxr-xr-x  3 utente2           1024 feb 10 11:15 viole
$ chmod g+w viole
$ ls -l viole
drwxrwxr-x  3 utente2           1024 feb 10 11:15 viole
$
```

Come si può osservare, per effetto del comando il trattino (`-`) nella sequenza di caratteri relativa alle autorizzazioni del gruppo è diventato una `w`.

Per eliminare dalla stessa directory le autorizzazioni di lettura ed esecuzione degli utenti esterni al gruppo (*autorizzazioni* viene sostituito in questo caso da `o-rx`), digitare il seguente comando:

```
$ ls -l viole
drwxrwxr-x  3 utente2           1024 feb 10 11:15 viole
$ chmod o-rx viole
$ ls -l viole
drwxrwx---  3 utente2           1024 feb 10 11:15 viole
$
```

In seguito al comando, la `r` (lettura) e la `x` (esecuzione) del gruppo di caratteri relativi alle autorizzazioni degli altri utenti sono state sostituite da trattini (`-`).

Quando si crea un nuovo file o una nuova directory, il sistema assegna automaticamente le autorizzazioni di default.

In generale, le autorizzazioni di default per i nuovi file sono le seguenti:

-rw-r--r--

Per le nuove directory sono invece le seguenti:

drwxr-xr-x

Perciò, per creare un nuovo file `petunia` che sia eseguibile dal proprietario (`utente2`) occorrerà procedere come segue:

```
$ ls -l petunia
-rw-r--r--  3 utente2           1024 feb 10 12:27 petunia
$ chmod u+x petunia
$ ls -l petunia
-rwxr--r--  3 utente2           1024 feb 10 12:27 petunia
$
```

Per modificare con un solo comando le autorizzazioni di tutte le categorie di utenti, usare l'opzione a. Per creare ad esempio un nuovo file rosa che sia eseguibile da tutti gli utenti, procedere come segue:

```
$ ls -l rosa
-rw-r--r--  3 utente2           1024 feb 10 11:31 rosa
$ chmod a+x rosa
$ ls -l rosa
-rwxr-xr-x  3 utente2           1024 feb 10 11:31 rosa
$
```

In seguito al comando, in tutte le categorie comparirà la lettera x.

È anche possibile cambiare le autorizzazioni di più file e directory utilizzando il carattere speciale *. Ad esempio, per cambiare le autorizzazioni di tutti i file della directory corrente fiori in modo che solo il proprietario vi abbia accesso di scrittura, procedere come segue:

```
$ pwd
/home/utente2/fiori
$ ls -l
-rwxrwxrwx  3 utente2          21032 feb 12 10:31 giglio
-rwxrwxrwx  3 utente2          12675 feb 08 09:31 mughetto
-rwxrwxrwx  2 utente2             68 feb 10 11:09 narciso
-rwxrwxrwx  1 utente2           1024 feb 14 16:38 rosa
$ chmod go-w *
$ ls -l
-rwxr-xr-x  3 utente2          21032 feb 12 10:31 giglio
-rwxr-xr-x  3 utente2          12675 feb 08 09:31 mughetto
-rwxr-xr-x  2 utente2             68 feb 10 11:09 narciso
-rwxr-xr-x  1 utente2           1024 feb 14 16:38 rosa
$
```

Nell'esempio, l'uso del comando pwd permette di verificare che la directory in cui viene eseguita l'operazione chmod sia la directory corrente.

Impostazione di autorizzazioni assolute

Finora si è parlato del comando `chmod` per la modifica delle autorizzazioni in modo *relativo* rispetto alle impostazioni correnti. Il comando `chmod` prevede tuttavia un'altra forma di utilizzo, in cui vengono impiegati codici numerici per specificare i diversi tipi di autorizzazioni. Utilizzato in questa forma, `chmod` permette di impostare le autorizzazioni di un file o di una directory in modo *assoluto*.

La sintassi di `chmod` per l'impostazione di autorizzazioni assolute è la seguente:

`chmod` *codnum nome*

dove *codnum* è il codice numerico e *nome* è il nome del file o della directory su cui si desidera eseguire l'operazione.

Il codice numerico completo è costituito da tre cifre. Ogni cifra viene utilizzata per una delle tre categorie di utenti: proprietario, gruppo e altri. Ad esempio, il comando seguente imposta le autorizzazioni assolute di lettura, scrittura ed esecuzione per il proprietario e per il gruppo e la sola autorizzazione di esecuzione per gli altri utenti:

```
$ chmod 771 rosa
```

La Tabella 3-1 illustra la correlazione tra le autorizzazioni descritte per il file `rosa` e il codice numerico `771`.

Tabella 3-1 Autorizzazioni del file `rosa`

Autorizzazione	**Proprietario**	**Gruppo**	**Altri**
Lettura	4	4	0
Scrittura	2	2	0
Esecuzione	1	1	1
Totale	7	7	1

Ogni colonna della Tabella 3-1 rappresenta una delle categorie di utenti: proprietario, gruppo e altri. Per impostare le autorizzazioni in lettura occorre aggiungere il codice 4 alla colonna appropriata; per impostare le autorizzazioni

in scrittura occorre aggiungere un 2, mentre per le autorizzazioni di esecuzione occorre aggiungere un 1. Il totale delle tre colonne indicato nell'ultima riga della tabella rappresenta il codice numerico completo.

Qui di seguito è illustrato un altro esempio dell'uso di questo metodo per impostare le autorizzazioni assolute, con l'aggiunta del comando `ls -l` per evidenziare il risultato ottenuto:

```
$ ls -l mughetto
-rw-r--r--  3 utente2          1024 feb 10 11:46 mughetto
$ chmod 755 mughetto
$ ls -l mughetto
-rwxr-xr-x  3 utente2          1024 feb 10 11:48 mughetto
$
```

Le autorizzazioni del file `mughetto` sono state impostate in modo che il proprietario possa leggere, scrivere ed eseguire il file, e in modo che i membri del gruppo e gli altri utenti possano leggerlo ed eseguirlo. La Tabella 3-2 illustra nei dettagli la composizione del codice numerico relativo alle autorizzazioni del file `mughetto`.

Tabella 3-2 Autorizzazioni del file `mughetto`

Autorizzazione	**Proprietario**	**Gruppo**	**Altri**
Lettura	4	4	4
Scrittura	2	0	0
Esecuzione	1	1	1
Totale	7	5	5

Per applicare al file `ciclamino` le autorizzazioni di lettura, scrittura ed esecuzione per tutte le categorie di utenti, il comando da eseguire sarà il seguente:

```
$ ls -l ciclamino
-rw-r--r--  3 utente2          1024 feb 10 11:51 ciclamino
$ chmod 777 ciclamino
$ ls -l ciclamino
-rwxrwxrwx  3 utente2          1024 feb 10 11:53 ciclamino
$
```

La Tabella 3-3 illustra la composizione del codice utilizzato per questo esempio.

Tabella 3-3 Autorizzazioni del file `ciclamino`

Autorizzazione	Proprietario	Gruppo	Altri
Lettura	4	4	4
Scrittura	2	2	2
Esecuzione	1	1	1
Totale	7	7	7

Il codice numerico `777` rappresenta il livello massimo di autorizzazioni che è possibile attribuire a un file o a una directory.

Come per la modifica delle autorizzazioni relative, anche nel caso delle autorizzazioni assolute è possibile usare il carattere speciale `*` per eseguire il comando su tutti i file della directory corrente. Ad esempio, per modificare le autorizzazioni assolute di tutti i file della directory corrente `fiori` in modo che il proprietario abbia le autorizzazioni di lettura, scrittura ed esecuzione, il gruppo le autorizzazioni di lettura ed esecuzione e gli altri utenti solo l'autorizzazione di esecuzione, inserire il comando seguente:

```
$ pwd
/home/utente2/fiori
$ ls -l
-rwxrwxrwx  3 utente2          21032 feb 12 10:31 giglio
-rwxrwxrwx  3 utente2          12675 feb 08 09:31 mughetto
-rwxrwxrwx  2 utente2             68 feb 10 11:09 narciso
-rwxrwxrwx  1 utente2           1024 feb 14 16:38 rosa
$ chmod 751 *
$ ls -l
-rwxr-x--x  3 utente2          21032 feb 12 10:31 giglio
-rwxr-x--x  3 utente2          12675 feb 08 09:31 mughetto
-rwxr-x--x  2 utente2             68 feb 10 11:09 narciso
-rwxr-x--x  1 utente2           1024 feb 14 16:38 rosa
$
```

Nell‘esempio viene utilizzato anche il comando `pwd` per verificare che la directory nella quale si esegue l’operazione sia la directory corrente. Il comando `ls -l` è inserito solo per illustrare il risultato della modifica delle autorizzazioni. Quando si impostano le autorizzazioni assolute non è necessario conoscere le autorizzazioni correnti.

Per maggiori informazioni sul comando `chmod`, vedere *man Pages(1): User Commands*.

Operazioni di ricerca 4

In questo capitolo viene descritto come ricercare all'interno di file o directory determinate stringhe o parole chiave utilizzando il comando `grep` di SunOS.

Ricerca di stringhe di caratteri con `grep`

Per ricercare una determinata stringa di caratteri in un file si utilizza il comando `grep`. La sintassi generale di `grep` è la seguente:

```
$ grep stringa file
```

dove *stringa* è la parola o la frase da ricercare e *file* è il nome del file nel quale deve essere eseguita la ricerca.

Nota – Una *stringa* è una sequenza di uno o più caratteri; può essere perciò una lettera singola, una parola o una frase. Le stringhe possono includere "spazi vuoti", segni di interpunzione e caratteri invisibili (di controllo).

Per ricercare ad esempio l'interno telefonico di Edgar Allan Poe, occorre digitare `grep`, il nome da ricercare in forma completa o parziale e il file che contiene l'informazione:

```
$ grep Poe interni
Edgar Allan Poe     x72836
$
```

Si noti che `grep` può individuare più righe corrispondenti al modello di ricerca impostato:

```
$ grep Allan interni
David Allan         x76438
Edgar Allan Poe     x72836
$ grep Al interni
Louisa May Alcott   x74236
David Allan         x76438
Edgar Allan Poe     x72836
$
```

`grep` non considera allo stesso modo le lettere maiuscole e le lettere minuscole; l'output del comando rispecchia perciò il modello di ricerca impostato anche nell'uso delle maiuscole:

```
$ grep allan interni
$ grep Allan interni
David Allan         x76438
Edgar Allan Poe     x72836
$
```

Con il primo comando dell'esempio `grep` non trova alcuna voce corrispondente al modello di ricerca, in quanto nessuno dei nomi presenti nel file inizia con una "a" minuscola.

Uso di `grep` *come filtro*

`grep` viene spesso utilizzato come "filtro" insieme ad altri comandi. Permette infatti di escludere le informazioni non richieste dall'output dei comandi. Per utilizzare `grep` come filtro occorre inviare a `grep` l'output del comando precedente utilizzando il simbolo pipe "`|`".

Il comando dell'esempio seguente ordina al sistema di elencare tutti i file che terminano in "`.ps`" creati nel mese di maggio (mag):

```
$ ls -l *.ps | grep mag
```

La prima parte della riga di comando,

```
ls -l *.ps
```

produce un elenco completo dei file:

```
$ ls -l *.ps
-rw-r--r--  1 elvis       7228 apr 22 15:07 bottoni.ps
-rw-r--r--  1 elvis       2356 mag 22 12:56 calendario.ps
-rw-r--r--  1 elvis       1567 giu 22 12:56 cmdtool.ps
-rw-r--r--  1 elvis      10198 giu 22 15:07 comando.ps
-rw-r--r--  1 elvis       5644 mag 22 15:07 orologio.ps
$
```

La seconda parte,

```
| grep mag
```

invia l'elenco ottenuto a `grep`, il quale ricerca il modello `mag`:

```
$ ls -l *.ps | grep mag
-rw-r--r--  1 elvis       2356 mag 22 12:56 calendario.ps
-rw-r--r--  1 elvis       5644 mag 22 15:07 orologio.ps
$
```

Uso di `grep` con stringhe di più parole

Per impostare un modello di ricerca formato da più parole separate, occorre racchiudere la stringa tra virgolette singole o doppie:

```
$ grep "Louisa May" interni
Louisa May Alcott     x74236
$
```

`grep` può ricercare una stringa anche in più file. Se il modello viene trovato in più file, `grep` stampa il nome del file seguito da due punti e dalla riga che contiene la stringa specificata:

```
$ grep Al *
alaska:l'Alaska è lo stato più grande degli USA.
attori:Alberto Sordi
poeti:Dante Alighieri.
$
```

Ricerca di righe che non contengono una data stringa

Per ricercare le righe di un file che *non* contengono una data stringa si utilizza l'opzione `-v` del comando `grep`. Nell'esempio seguente, vengono ricercate tutte le righe dei file della directory home dell'utente `medici` che non contengono la lettera `e`:

```
$ ls
alaska    attori    hinterland    insegnanti    poeti
$ grep -v e *
attori:sabato 14 marzo 1993, 10:00 MET
poeti:Guido Gozzano.
$
```

Altri usi di `grep`

Il comando `grep` può anche essere utilizzato con le cosiddette *espressioni regolari*. Le espressioni regolari sono formate da lettere o numeri abbinati a caratteri speciali che `grep` interpreta in modo particolare. Questi caratteri

speciali, chiamati anche *metacaratteri*, hanno un significato particolare anche per il sistema; è perciò necessario digitarli tra virgolette o farli precedere da un carattere di escape. Per utilizzare `grep` con un'espressione regolare al prompt di sistema, inserire l'espressione tra virgolette o utilizzare i metacaratteri (`& ! . * $ ?` e `\`) preceduti da una barra rovesciata (\).

- L'accento circonflesso (^) indica l'inizio di una riga. Perciò il comando:

```
$ grep '^b' lista
```

  cercherà tutte le righe del file `lista` che iniziano con "b."

- Il simbolo del dollaro ($) indica la fine di una riga. Il comando:

```
$ grep 'b$' lista
```

  visualizzerà tutte le righe del file `lista` in cui l'ultimo carattere è una "b". Il comando:

```
$ grep '^b$' lista
```

  visualizzerà tutte le righe del file `lista` che contengono *solo* il carattere "b".

- All'interno di un'espressione regolare, un punto (`.`) sostituisce qualsiasi carattere singolo. Ad esempio, il comando:

```
$ grep 'de.' lista
```

  estrae dal file `lista` tutti i gruppi di tre caratteri in cui "de" siano i primi due, ad esempio "dei", "del", "scodella" e "code" (anche gli spazi vuoti sono considerati caratteri).

- Se un carattere è seguito da un asterisco (`*`), `grep` lo interpreta come "nessuna o più ricorrenze di quel carattere". Se l'asterisco è posto dopo un'espressione regolare, `grep` lo interpreta come "nessuna o più ricorrenze dei caratteri corrispondenti al modello di ricerca".

Il fatto che grep possa individuare anche i casi di “nessuna ricorrenza” di un carattere o di un modello di ricerca rende l’uso dell’asterisco poco intuitivo. Volendo ad esempio ricercare tutte le parole che contengano le lettere “qu”, il comando:

```
$ grep 'qu*' lista
```

produrrà l’effetto desiderato. Se tuttavia si vogliono ricercare tutte le parole che contengano la lettera “n”, il comando dovrà avere la forma seguente:

```
$ grep 'nn*' lista
```

Per ricercare tutte le parole che contengano il modello “nn”, il comando da inserire sarà il seguente:

```
$ grep 'nnn*' lista
```

Per comprendere meglio questa funzione di grep può essere opportuno effettuare alcune prove ed esaminare l’output dei comandi.

- Per ricercare nessuna o più ricorrenze di un *qualsiasi* carattere del file lista, digitare:

```
$ grep .* lista
```

Ricerca di metacaratteri

Si supponga di voler ricercare in un testo le righe che contengano il simbolo del dollaro ($). In questo caso, nell’espressione regolare il simbolo del dollaro deve essere preceduto da una barra rovesciata (\), in modo che grep ignori il significato speciale del carattere. Lo stesso vale anche per gli altri metacaratteri (& ! . * ? e \).

Ad esempio, l’espressione

```
$ grep ^\.
```

individua tutte le righe che iniziano con un punto, ed è particolarmente utile nella ricerca di richieste di formattazione `nroff` o `troff` (che iniziano con un punto).

La Tabella 4-1 contiene un elenco degli elementi più usati nei modelli di ricerca impostati con `grep`.

Tabella 4-1 Elementi speciali utilizzati nei modelli di ricerca con `grep`

Carattere	Per ricercare
^	L'inizio di una riga di testo
$	La fine di una riga di testo
.	Un carattere singolo
[...]	Un carattere singolo nell'elenco o nell'intervallo specificato tra parentesi
[^...]	un carattere non compreso nella lista o nell'intervallo specificato
*	Nessuna o più ricorrenze del carattere o dell'espressione regolare precedente
.*	Nessuna o più ricorrenze di un singolo carattere
\	Esclude il significato speciale del carattere successivo

Questi parametri di ricerca possono essere usati anche nell'editor di testo `vi`.

Virgolette singole e doppie nelle righe di comando

Come già accennato in precedenza, le virgolette vengono utilizzate per racchiudere un testo che deve essere interpretato come un'unica stringa. Ad esempio, per ricercare con `grep` in tutti i file della directory corrente l'espressione “luna piena”, il comando da utilizzare è il seguente:

```
$ grep "luna piena" *
```

Per riunire più parole in una singola unità è anche possibile utilizzare le virgolette singole ('). Questo tipo di virgolette garantisce inoltre che determinati caratteri, ad esempio $, vengano interpretati in modo letterale. Si noti tuttavia che il metacarattere ! utilizzato con history viene sempre interpretato come tale, anche racchiuso tra virgolette, a meno che non sia preceduto da una barra rovesciata. In ogni caso, per utilizzare in modo letterale caratteri speciali come & ! $? . ; e \, è buona norma farli precedere da un carattere di escape.

Ad esempio, digitando il comando:

```
$ grep $ lista
```

verranno visualizzate *tutte* le righe del file lista. Digitando invece:

```
$ grep '\$' lista
```

verranno visualizzate solo le righe che contengono il carattere “$”.

Per maggiori informazioni sul comando grep, vedere *man Pages(1): User Commands*.

Password, processi e spazio sul disco

5

SunOS dispone di una serie di comandi per l'esecuzione delle operazioni di sistema dalla riga di comando. In questo capitolo vengono descritte le procedure da seguire per impostare una password, elencare i processi eseguiti sul sistema, interrompere un processo e visualizzare lo spazio occupato sul disco.

Utilizzo di una password

Per garantire la sicurezza del sistema, SunOS richiede l'inserimento di una password di accesso. Il cambiamento frequente della password rappresenta una misura di sicurezza aggiuntiva per escludere l'accesso al sistema da parte di persone non autorizzate. Se si sospetta che qualcuno abbia usato il proprio profilo utente senza autorizzazione, è buona norma cambiare immediatamente la password.

Per la scelta della password è consigliabile osservare i seguenti criteri:

- Scegliere una password che si riesca a ricordare a memoria. Una password difficile da ricordare è peggio di una password troppo facile da indovinare.
- Scegliere una password formata da almeno sei caratteri e che contenga almeno un numero.
- Non utilizzare il proprio nome o le proprie iniziali né il nome o le iniziali di amici o parenti.
- Non utilizzare nomi di animali o di oggetti legati ai propri interessi.

- Non utilizzare solo lettere maiuscole.
- Se si dispone di più profili utente, non utilizzare la stessa password per tutti.
- Benché in linea generale sia ammesso l'utilizzo di qualsiasi carattere, è preferibile evitare alcuni caratteri o combinazioni come Ctrl-C, Ctrl-Z, Ctrl-U, Ctrl-S, Esc, Tab e, in alcuni casi, # e @, che potrebbero essere interpretati dal terminale come segnali. In questo caso, l'interpretazione errata del carattere impedirebbe di inserire la password correttamente.

Cambiamento della password

Per cambiare la propria password, digitare il comando `passwd`:

```
$ passwd
Changing password for hankw on worker
Old password:
New password:
Retype new password:
$
```

1. **Al prompt** `Old Password:`**, digitare la password corrente.**
   Se al profilo utente non è ancora assegnata una password, il sistema non presenterà questo prompt. Si noti che il sistema non visualizza la password sullo schermo mentre la si inserisce. Si tratta di un'ulteriore misura di sicurezza per nascondere la propria password ad altri utenti.

2. **Al prompt** `New Password:`**, inserire la nuova password prescelta.**
   Anche in questo caso, la password inserita non verrà visualizzata sullo schermo.

3. **Al prompt finale,** `Retype new password:`**, inserire per la seconda volta la nuova password.**
   Questo passaggio permette di verificare di avere inserito esattamente la password desiderata.

   Se la nuova password non viene inserita esattamente nella stessa forma ad entrambe le richieste, il sistema non effettua il cambiamento e risponde con il messaggio `Sorry`. Se questo accade più volte, richiedere una nuova password all'amministratore del sistema.

Nota – Non sono ammesse password che contengano meno di sei caratteri. Inoltre, ogni nuova password deve contenere almeno tre caratteri diversi dalla vecchia.

Durata della password

Se è stata attivata la funzione di “aging” (durata) della password, implementata con speciali opzioni del comando `passwd`, è possibile che la password abbia una durata massima, oppure una durata massima *e* una durata minima. La durata della password viene impostata dall’amministratore del sistema.

Al raggiungimento della data di scadenza (o durata massima) della password, il sistema richiede di impostarne una nuova. La richiesta viene presentata al login con il messaggio seguente:

```
Your password has expired. Choose a new one.
```

A questo punto il sistema esegue automaticamente il comando `passwd` e richiede una nuova password.

Se ad esempio la durata *minima* della password è stata fissata a due settimane e si cerca di cambiarla prima di tale scadenza, il sistema visualizza il messaggio seguente:

```
Sorry, less than 2 weeks since the last change.
```

Per visualizzare le informazioni relative alla durata della password, usare il comando `passwd` con l’opzione `-d`:

```
$ passwd -d
nomeutente 2-14-92 14 60
```

Vengono visualizzate, nell’ordine, la data di creazione, la durata minima e la durata massima della password corrente. (Queste informazioni compaiono solo se è stata implementata la funzione di aging della password.)

Per maggiori informazioni sulle password e sulle opzioni di aging, vedere *man Pages(1): User Commands*.

Processi e PID

Non appena il sistema interpreta un comando, per la sua esecuzione viene creato un *processo* indipendente dotato di un numero di identificazione (PID) esclusivo. Il sistema utilizza il PID per tenere traccia dello stato corrente di ogni processo.

Visualizzazione dei comandi in esecuzione (`ps`)

Il comando `ps` permette di visualizzare i processi in corso di esecuzione. Oltre a visualizzare il *numero di identificazione del processo* (`PID`) per tutti i processi di proprietà dall'utente (cioè creati in seguito a un comando digitato dall'utente), `ps` indica anche il *terminale* dal quale il processo è stato avviato (`TTY`), il *tempo della cpu* utilizzato fino a quel momento (`TIME`) e il *comando* in corso di esecuzione (`COMMAND`).

Aggiungendo l'opzione `-l` al comando `ps`, il sistema visualizza una serie di altre informazioni relative ai processi correntemente attivi, tra cui lo *stato* di tutti i processi (elencati sotto `S`). I codici utilizzati per l'indicazione degli stati sono i seguenti:

- `O` - Il processo è attivo su un processore.
- `S` - *Sleeping*: il processo sta attendendo la conclusione di un evento.
- `R` - *Runnable*: il processo è nella coda di esecuzione.
- `I` - *Idle*: il processo è in fase di creazione.
- `Z` - Stato *Zombie*: il processo è terminato e il processo padre non è in attesa.
- `T` - *Traced*: il processo è stato interrotto da un segnale perché il processo padre sta eseguendo un `trace` su di esso.
- `X` - Stato *SXBRK*: il processo attende che si renda disponibile una maggiore quantità di memoria centrale.

È importante ricordare che durante l'esecuzione di ps lo stato dei processi può cambiare. Poiché il comando ps fornisce solo uno spaccato di ciò che accade in un determinato momento, le informazioni visualizzate valgono solo per l'istante successivo all'invio del comando. Di conseguenza, l'output di ps potrebbe non essere perfettamente aggiornato.

Il comando ps dispone di altre opzioni non trattate in questo paragrafo. Per informazioni più approfondite, vedere *man Pages(1): User Commands*.

Interruzione dei processi (kill)

Il comando kill permette di interrompere in modo diretto un processo in esecuzione. Questo risulta particolarmente utile quando si avvia per errore un comando che comporta un certo tempo di esecuzione.

Per interrompere un processo:

1. **Digitare ps per visualizzare il PID del processo.**
2. **Digitare kill seguito dal PID del processo.**

L'esempio seguente illustra la procedura da seguire:

```
$ ps
PID    TTY    TIME    COMMAND
1291   co     0:12    -bin/csh (csh)
3250   p0     0:00    ps
1286   p1     0:05    -bin/csh (csh)
3248   p1     0:05    vi commands
$ kill 1291
[1}  Terminated       -bin/csh/ (csh)
$
```

Un metodo più rapido per determinare il PID del processo consiste nell'inviare l'output di ps al comando grep utilizzando il simbolo di pipe. Ad esempio:

```
$ ps | grep nomecomando
```

dove *nomecomando* è il nome del processo che si desidera interrompere. (L'uso del comando grep è descritto al Capitolo 4, "Operazioni di ricerca".)

Per interrompere forzatamente un processo, usare il comando `ps` con l'opzione `-9` nella forma seguente:

```
$ kill -9 nPID
```

dove *nPID* è il numero di identificazione del processo che si desidera interrompere.

Gestione dello spazio sul disco

Poiché lo spazio sul disco è una risorsa limitata, è buona norma tenersi aggiornati sullo stato di occupazione del proprio sistema.

Visualizzazione dello spazio occupato sul disco (`df -k`)

Il comando `df -k` visualizza lo spazio correntemente occupato in ogni disco accessibile (attivato) del proprio sistema. Digitando:

```
$ df -k
```

vengono visualizzati la capacità di ogni disco attivato sul sistema, lo spazio disponibile e la percentuale di spazio già occupata.

Se un file system risulta occupato al 90% o oltre è consigliabile cancellare i file non necessari. È possibile ad esempio spostare i file su un altro disco o su un nastro usando `cp` per copiarli e quindi `rm` per cancellarli, oppure cancellarli direttamente. Naturalmente si raccomanda di eseguire queste operazioni di "manutenzione" solo sui propri file.

Visualizzazione dello spazio occupato dalle directory (`du`)

Il comando `du` permette di visualizzare lo spazio occupato da una directory e dalle relative sottodirectory in blocchi di 512 byte, vale a dire in unità di 512 byte o caratteri.

du visualizza lo spazio occupato da ogni sottodirectory. Per ottenere un elenco delle sottodirectory di un file system, spostarsi con cd nella directory superiore del file system e digitare il seguente comando:

```
$ du | sort -r -n
```

Questo comando, che utilizza le opzioni di ordinamento inverso (-r) e numerico (-n) del comando sort, permette di identificare facilmente le directory di grandi dimensioni. Con il comando ls -l è quindi possibile esaminare le dimensioni (in byte) e le date di modifica dei file contenuti nelle singole directory. L'eccessiva occupazione dello spazio sul disco è spesso dovuta alla presenza di vecchi file o di file di testo che occupano più di 100 Kbyte.

L'editor `vi` 6

`vi` (pronunciato “vi-ai”) è l'editor di testo standard di SunOS. Poiché non richiede un'interfaccia a finestre, `vi` può essere utilizzato su qualsiasi tipo di terminale per creare o modificare vari tipi di file.

`vi` può essere utilizzato per inserire e modificare un testo, ma non può essere definito un vero e proprio word-processor. A differenza dei word-processor disponibili in commercio, infatti, `vi` non è stato creato per elaborare testi formattati. Per produrre stampe formattate `vi` si affida a programmi in grado di emulare un processo di composizione come `nroff`, `troff` o `ditroff`. Questi programmi consentono di formattare il testo di `vi` inserendo una serie di codici che vengono successivamente interpretati dall'emulatore.

La lunga serie dei comandi di `vi`, molti dei quali hanno funzioni simili, può creare qualche confusione nei nuovi utenti. In questo capitolo viene fornita una panoramica dei comandi fondamentali. Si scoprirà con l'uso che `vi` è un editor di testo estremamente potente, ma anche che per impiegarlo con efficacia è indispensabile una certa pratica.

Esiste anche una versione di `vi` di sola lettura, di nome `view`. Aprendo un file con `view` si possono usare gli stessi comandi di `vi`, ma non è possibile salvare le modifiche. Questo programma può essere utilizzato quando occorre leggere un file di `vi` senza rischiare di modificarlo accidentalmente.

Avvio di `vi`

Nei paragrafi seguenti verrà spiegato come avviare `vi`, inserire un testo in un file, salvare (scrivere) il file e uscire da `vi`. Per facilitare la comprensione delle operazioni descritte, la spiegazione è accompagnata da esempi pratici.

Creazione di un file

Avviare `vi` aprendo il file `prova` come mostrato in questo esempio:

```
$ vi prova
```

Se esiste già un file di nome `prova`, `vi` aprirà il file esistente; in caso contrario verrà creato un nuovo file. Nell'esempio qui descritto, `prova` è un nuovo file.

Compare la videata di modifica di `vi`:
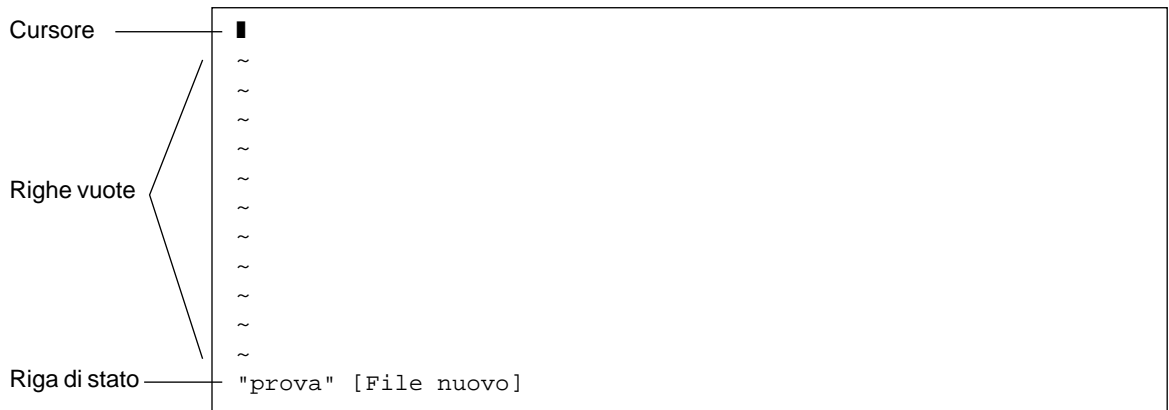
Figura 6-1 La videata di modifica di `vi`

Il cursore appare nell'angolo superiore sinistro dello schermo. Le righe vuote sono indicate da una tilde (~).

`vi` può essere avviato anche senza specificare il nome di un file, digitando semplicemente `vi`. Il file creato potrà essere denominato all'uscita da `vi`.

La riga di stato

L’ultima riga dello schermo, detta *riga di stato*, presenta il nome del file e il numero di righe e di caratteri che esso contiene. Quando si crea un nuovo file, come nell’esempio precedente, la riga di stato segnala che si tratta di un file nuovo.

Le due modalità di `vi`

`vi` dispone di due modalità operative: la *modalità inserimento*, utilizzata per inserire un testo in un file, e la *modalità comando*, usata per inserire comandi che eseguono specifiche funzioni. La modalità di default è la modalità comando.

Poiché `vi` non indica la modalità corrente, la distinzione tra modalità inserimento e modalità comando causa spesso una certa confusione tra i nuovi utenti dell’editor. Per evitare di incorrere in uno “stress da vi”, sarà sufficiente avere chiari sin dall’inizio alcuni concetti fondamentali.

Quando si apre un file in `vi`, l’editor è sempre in modalità comando. Per poter inserire un testo occorre prima digitare uno dei comandi che abilitano la modalità di inserimento, ad esempio `i` (“insert”), per inserire il testo *nella* posizione corrente del cursore, oppure `a` (“append”), per inserire il testo *dopo* la posizione corrente del cursore. (Questi ed altri comandi di `vi` verranno trattati in modo più approfondito nel resto del capitolo.)

Per tornare alla modalità di comando è sufficiente premere Esc. Se non si è certi della modalità corrente di `vi`, è possibile premere Esc per entrare in modalità comando e avere così un punto di riferimento sicuro. Se si preme Esc quando `vi` si trova già in modalità comando, il sistema emette un segnale acustico e lo schermo lampeggia, ma non si produce alcun danno.

Modalità inserimento

Per inserire un testo nel file `prova`, digitare il comando di inserimento `i`. `vi` esce dalla modalità comando ed entra in modalità inserimento.

Inserire ora alcune righe di testo terminando ogni riga con un Return. Come si può osservare, i caratteri digitati appaiono a sinistra del cursore e spingono i caratteri già presenti verso destra. Per il momento, se si commettono errori di

battitura tornare indietro con il tasto Backspace e digitare nuovamente la riga prima di premere Return. Più avanti verrà spiegato come modificare il testo inserito.

Dopo avere inserito alcune righe, premere Esc per tornare alla modalità comando. Il cursore si sposterà sull'ultimo carattere inserito. A questo punto è possibile inserire altri comandi di `vi`.

Se `vi` sembra rispondere in modo strano, verificare che il tasto "Caps Lock" non sia premuto; l'attivazione di questo tasto fa apparire tutto il testo inserito in lettere maiuscole. In alcuni sistemi il tasto F1 (di norma posizionato vicino al tasto Esc) ha la stessa funzione del tasto Caps Lock. Si faccia attenzione, in questi casi, a non premere erroneamente questo tasto al posto di Esc.

Nota – Talvolta può essere necessario azzerare o ridisegnare lo schermo, ad esempio per eliminare un messaggio del sistema. Per ridisegnare lo schermo, entrare in modalità comando e premere Ctrl-L. L'effetto di questo comando è analogo a quello dell'opzione "Ridisegna schermo" di OpenWindows.

Modalità comando

Quando si apre un file, `vi` si trova sempre in modalità comando. In questa modalità è possibile inserire una serie di comandi associati ad altrettante funzioni. Quasi tutti i comandi di `vi` consistono in una o due lettere, talvolta accompagnate da un numero opzionale. Di norma esistono due versioni dello stesso comando, una a lettere maiuscole e l'altra a lettere minuscole, che eseguono funzioni diverse ma correlate. Ad esempio, digitando una `a` minuscola il testo viene inserito a destra del cursore, mentre digitando una `A` maiuscola il testo viene inserito alla *fine* della riga.

In genere `vi` esegue i comandi direttamente, senza bisogno di premere Return. Un'eccezione è rappresentata dai comandi che iniziano con i due punti (:). In alcune guide di `vi` questi comandi vengono classificati come una terza modalità operativa, detta *modalità ultima riga*. Digitando due punti in modalità comando, infatti, i due punti e i caratteri inseriti successivamente appaiono sull'ultima riga dello schermo. Nell'ambito di questa descrizione, tuttavia, tutti i comandi di `vi` verranno considerati inseriti in modalità comando.

I comandi preceduti dai due punti sono in realtà comandi di *ex*. `vi` ed `ex` sono due interfacce separate dello stesso programma di trattamento testi, ma mentre `vi` è un'interfaccia "a schermo", `ex` è un'interfaccia "a linee". Tutti i comandi di `ex` sono disponibili anche in `vi`. In particolare, premendo il tasto dei due punti si passa direttamente all'interfaccia "a linee" `ex`. Questa speciale modalità consente di eseguire molte operazioni sui file senza bisogno di uscire da `vi`. Per maggiori informazioni, vedere "Uso dei comandi di ex" in questo capitolo.

Termine di una sessione

Quando si modifica un file in `vi`, in un primo momento le modifiche non vengono apportate direttamente al file, bensì a una copia del file che `vi` crea in uno spazio di memoria temporaneo detto *buffer*. La copia permanente del file sul disco viene modificata solo al momento della *scrittura* (salvataggio) del contenuto del buffer.

Questa procedura presenta alcuni aspetti positivi e alcuni negativi. Da un lato permette di uscire da un file e abbandonare le modifiche effettuate durante una sessione lasciando intatta la copia sul disco. Dall'altro lato vi è il rischio di perdere il contenuto del buffer (non salvato) in caso di crash del sistema. (Gli utenti collegati a terminali remoti tramite linea telefonica sono particolarmente soggetti a questo tipo di interruzioni.)

Il metodo migliore consiste nel salvare il lavoro frequentemente, soprattutto quando vengono effettuate modifiche di una certa consistenza.

Attenzione – `vi` permette di tenere aperte più copie di uno stesso file simultaneamente. Si consiglia tuttavia di limitare il più possibile l'uso di questa funzione, che potrebbe creare confusione tra le modifiche apportate al file corrente e quelle inserite in una sessione simultanea.

Salvataggio delle modifiche e uscita da `vi`

`vi` dispone di diversi comandi per il salvataggio del contenuto del buffer e l'uscita dall'editor. Questi comandi offrono la possibilità di salvare il file, salvare e uscire oppure uscire senza salvare.

Salvataggio

Per salvare il contenuto del buffer (cioè scrivere il contenuto del buffer nel file sul disco) digitare:

```
:w
```

seguito da Return.

Salvataggio e uscita

Per salvare e uscire dall'editor, digitare:

```
:wq
```

seguito da Return. Oppure, digitare `ZZ`.

Si noti che il comando `ZZ` non è preceduto dai due punti e non deve essere seguito da Return.

Uscita senza salvataggio

Se il file non è stato modificato e si desidera semplicemente uscire, digitare:

```
:q
```

seguito da Return. Se il file è stato modificato, `vi` non permetterà di uscire con `:q`. Verrà visualizzato il messaggio `Le ultime modifiche non sono state salvate (:quit! per ignorare).`

Se non si desidera salvare le modifiche, digitare:

```
:q!
```

seguito da Return.

Stampa di un file

Dopo avere chiuso il file da `vi`, è possibile stamparlo con il comando seguente:

```
$ lp nomefile
```

dove *nomefile* è il nome del file da stampare. Questo comando invia il file alla stampante di default. Il file viene stampato senza alcuna formattazione, riga per riga, esattamente nella forma in cui appare sullo schermo. Per maggiori informazioni sui comandi di stampa, vedere il Capitolo 8, "Uso delle stampanti".

Comandi base di `vi`

I paragrafi seguenti illustrano diverse categorie di comandi di `vi`, utilizzati tra l'altro per le seguenti operazioni:

- Spostamento all'interno di un file
- Inserimento di un testo
- Modifica e sostituzione di un testo
- Annullamento delle modifiche apportate a un testo
- Cancellazione di un testo
- Copia e spostamento di un testo
- Ripetizione di comandi

Spostamento all'interno di un file

Nei paragrafi precedenti è stato spiegato come creare, salvare, stampare e chiudere un file di `vi`. A questo punto, avendo creato un file, è necessario comprendere i concetti essenziali per gli spostamenti al suo interno. Aprire il file di prova ed eseguire i comandi descritti qui di seguito.

Spostamento del cursore

All'avvio di `vi`, il cursore compare nell'angolo superiore sinistro dello schermo. Operando in modalità comando è possibile spostare il cursore tramite una serie di comandi da tastiera, associati caso per caso a determinate lettere, ai tasti freccia, al tasto Return, al tasto Back Space (o Canc) o alla barra spaziatrice.

Nota – vi interpreta i comandi distinguendo tra maiuscole e minuscole; uno "stesso" comando scritto in lettere maiuscole e in lettere minuscole potrebbe perciò produrre effetti radicalmente diversi.

Spostamenti con i tasti freccia

Se la tastiera dispone dei tasti freccia, si provi ad utilizzarli per spostare il cursore. Questi tasti permettono di muovere liberamente il cursore in alto, in basso, a destra e a sinistra. Si noti tuttavia che il cursore può essere spostato solo all'interno di un testo o di spazi già esistenti.

Se si utilizza vi da un terminale remoto, è possibile che i tasti freccia non funzionino correttamente. I problemi di questo tipo dipendono in genere dall'emulatore del terminale. Se i tasti freccia non sono utilizzabili, sono disponibili le seguenti alternative:

- Per lo spostamento a sinistra, premere h.
- Per lo spostamento a destra, premere l.
- Per lo spostamento in basso, premere j.
- Per lo spostamento in alto, premere k.

Spostamento di parola in parola

Premere w ("word") per spostare il cursore di parola in parola verso destra.

Premere b ("back") per spostare il cursore di parola in parola verso sinistra.

Premere W o B per spostare il cursore alla parola successiva o precedente ignorando i segni di punteggiatura.

Premere e ("end") per spostare il cursore sull'ultimo carattere della parola corrente.

Spostamento all'inizio o alla fine della riga

Premere ^ per spostare il cursore all'inizio della riga corrente.

Premere $ per spostare il cursore alla fine della riga corrente.

Spostamento alla riga successiva

Premere il tasto Return per spostare il cursore all'inizio della riga successiva.

Spostamento a sinistra

Premere il tasto Back Space per spostare il cursore di un carattere verso sinistra.

Spostamento a destra

Premere la barra spaziatrice per spostare il cursore di un carattere verso destra.

Spostamento a inizio schermo

Premere `H` ("high") per spostare il cursore all'inizio dello schermo.

Spostamento a metà schermo

Premere `M` ("middle") per spostare il cursore a metà schermo.

Spostamento a fine schermo

Premere `L` ("low") per spostare il cursore al fondo dello schermo.

Spostamento di pagina in pagina e scorrimento

Se ci si sposta verso il basso quando il cursore si trova al fondo dello schermo, o verso l'alto quando il cursore si trova all'inizio dello schermo, il testo scorrerà rispettivamente verso l'alto o verso il basso. Questo metodo può essere utile per lo spostamento all'interno di un file breve, ma può risultare scomodo o lento in un file di grandi dimensioni.

Gli spostamenti all'interno di un file possono essere eseguiti di pagina in pagina, di videata in videata o di mezza videata alla volta. (Per provare questi comandi sul file `prova`, aggiungere nuove righe per ottenere un testo di lunghezza sufficiente.)

Si noti che esiste una differenza fondamentale tra spostamento di pagina in pagina e scorrimento. Lo scorrimento produce un avanzamento o un arretramento di *una riga alla volta*, come se il testo venisse visualizzato su un rullo continuo. Lo spostamento di pagina in pagina fa invece avanzare o arretrare il cursore di *una videata alla volta*. Sui sistemi veloci la differenza può essere trascurabile. Se viceversa si lavora su un terminale remoto o in altre situazioni in cui il sistema opera in modo più lento del normale, la differenza può essere determinante.

Spostamento di una videata in avanti

Per spostarsi di una videata in avanti (verso il basso), premere Ctrl-F. (Premere il tasto Control e contemporaneamente il tasto F.) Il cursore apparirà nell'angolo superiore sinistro della nuova videata.

Spostamento di mezza videata in avanti

Per spostarsi di mezza videata in avanti, premere Ctrl-D.

Spostamento di una videata all'indietro

Per spostarsi di una videata all'indietro (verso l'alto), premere Ctrl-B.

Spostamento di mezza videata all'indietro

Per spostarsi di mezza videata all'indietro, premere Ctrl-U.

Inserimento di un testo

Nei paragrafi seguenti verranno descritti i comandi fondamentali di `vi` per l'inserimento di un testo. Si noti che tutti questi comandi pongono `vi` in modalità inserimento e che, per utilizzarli, occorre trovarsi in modalità comando. Per verificare di trovarsi in modalità comando, premere Esc.

Aggiunta di un testo

Digitando una `a` (da *append*, aggiungere), il nuovo testo viene inserito a *destra* del cursore. Per provare il funzionamento del comando, posizionare il cursore all'interno di una riga, premere una a minuscola e inserire il nuovo testo. Al termine, premere Esc.

Digitando una `A` maiuscola, il testo viene aggiunto alla *fine* della riga. Per provare il funzionamento di questo comando, posizionare il cursore all'interno di una riga e digitare `A`. Il cursore si sposterà alla fine della riga. Inserire il nuovo testo. Al termine, premere Esc.

Inserimento di un testo

Digitando una `i` minuscola in modalità comando, il nuovo testo viene inserito a sinistra del cursore.

Digitando una `I` maiuscola, il testo viene inserito all'inizio della riga. (Il comando fa spostare automaticamente il cursore all'inizio della riga.) Per tornare in modalità comando dopo avere inserito il testo, premere Esc.

Creazione di una riga

`vi` dispone di comandi per creare nuove righe sopra o sotto la posizione corrente del cursore.

Digitare una `o` minuscola per creare una riga *sotto* la posizione corrente del cursore. Per provare il comando, digitare una `o` e inserire un testo. Il nuovo testo inserito può occupare anche diverse righe. Al termine, premere Esc.

Digitare una `O` maiuscola per creare una riga *sopra* la posizione corrente del cursore.

Modifica di un testo

La modifica di un testo comporta la sostituzione di caratteri esistenti con caratteri nuovi. `vi` dispone di molti comandi per eseguire questa operazione, a seconda delle esigenze e della portata delle modifiche.

Modifica di una parola

Per sostituire una parola, posizionare il cursore all'inizio della parola da sostituire. Digitare `cw` seguito dalla nuova parola. Per terminare, premere Esc.

Per modificare una *parte* di una parola, posizionare il cursore sulla parola, a *destra* della parte da conservare. Digitare `cw`, inserire la correzione e premere Esc.

Modifica di una riga

Per modificare una riga, posizionare il cursore in qualsiasi punto della riga e digitare `cc`. La riga scomparirà lasciando una riga vuota per il nuovo testo (che potrà avere qualsiasi lunghezza). Per terminare, premere Esc.

Modifica di parte di una riga

Per sostituire una parte di una riga, posizionare il cursore a *destra* della parte che si desidera conservare. Digitare `C`, inserire il nuovo testo e premere Esc. In questo modo viene modificata la parte di testo compresa tra la posizione corrente del cursore e la fine della riga.

Sostituzione di caratteri

Per sostituire il carattere in corrispondenza del cursore con uno o più caratteri, digitare una `s` seguita dal nuovo testo. Per tornare alla modalità comando, premere Esc.

Sostituzione di un singolo carattere

Questo comando permette di sostituire il carattere evidenziato dal cursore con un altro carattere. Posizionare il cursore sul carattere e digitare una `r` seguita da un solo carattere sostitutivo. Dopo la sostituzione, `vi` torna automaticamente alla modalità comando (senza bisogno di premere Esc).

Correzione di caratteri trasposti

`vi` dispone di un comando molto semplice per correggere i caratteri trasposti, ad esempio un “`ter`” digitato al posto di “`tre`”. Per eseguire la correzione, posizionare il cursore sulla prima lettera da spostare (in questo caso la `e`), quindi digitare `xp`. La `e` e la `r` cambieranno di posto e `vi` tornerà automaticamente alla modalità comando.

Divisione o unione di righe

Per spezzare una riga senza modificarne il testo, portare il cursore su un carattere di spazio e digitare una `r` seguita da Return. Si noti che eseguendo questa operazione con il cursore posizionato su una lettera, la lettera evidenziata sarà sostituita dal carattere di ritorno a capo.

Per unire due righe, posizionare il cursore sulla riga superiore e digitare una `J` maiuscola. (Non è necessario premere Esc per terminare.)

Annullamento delle modifiche

Lavorando su un file di testo, capita spesso di effettuare una modifica e di accorgersi subito dopo di preferire la versione precedente. I comandi di annullamento di `vi` permettono di annullare gli effetti di un'operazione e di proseguire da quel punto.

Annullamento del comando precedente

Se si commette un errore o si cambia idea dopo avere eseguito una data operazione, è possibile annullare l'ultimo comando digitando immediatamente una `u`. (Non è necessario premere Esc per terminare.) Premendo `u` una *seconda* volta, `vi` annullerà il comando di annullamento.

Annullamento delle modifiche a una riga

Per annullare tutte le modifiche apportate a una riga, digitare una `U` maiuscola. Questo comando ha effetto solo se il cursore non è stato precedentemente spostato su un'altra riga. (Non è necessario premere Esc per terminare.)

Cancellazione di un testo

I comandi di `vi` descritti qui di seguito permettono di cancellare un carattere, una parola o una riga. Al termine di queste operazioni `vi` resta in modalità comando, perciò per inserire nuovo testo occorre prima entrare in modalità inserimento.

Cancellazione di un carattere

Per cancellare un carattere, posizionare il cursore sul carattere da cancellare e digitare `x`.

Il comando `x` cancella anche lo spazio occupato dal carattere — se si cancella una lettera all'interno di una parola, le altre lettere si avvicinano senza lasciare spazi vuoti. Il comando `x` permette anche di cancellare spazi vuoti all'interno di una riga.

Per cancellare un carattere posto prima (a sinistra) del cursore, digitare una `X` maiuscola.

Cancellazione di una parola o di parte di una parola

Per cancellare una parola, posizionare il cursore sul suo carattere iniziale e digitare `dw`. La parola viene cancellata senza lasciare spazi vuoti.

Per cancellare una parte di una parola, portare il cursore a *destra* della parte che si desidera conservare, quindi digitare `dw` per cancellare il resto della parola.

Cancellazione di una riga

Per cancellare una riga, posizionare il cursore in qualsiasi punto della riga e digitare `dd`. La riga viene cancellata senza lasciare spazi vuoti.

Cancellazione di parte di una riga

È anche possibile cancellare solo una parte di una riga.

Per cancellare tutti i caratteri situati a *destra* del cursore, posizionare il cursore a destra della parte che si desidera conservare e digitare `D`.

Per cancellare tutti i caratteri situati a *sinistra* del cursore, posizionare il cursore a destra della parte che si desidera cancellare e digitare `d0` (`d`-zero).

Cancellazione fino alla fine del file

Per cancellare tutto il testo compreso tra la riga corrente e la fine del file, digitare `dG`. Verrà cancellata anche la riga su cui è posizionato il cursore.

Cancellazione dall'inizio del file

Per cancellare tutto il testo compreso tra l'inizio del file e la riga corrente, digitare `d1G`. Verrà cancellata anche la riga su cui è posizionato il cursore.

Copia e spostamento di un testo — yank, delete e put

Molti word-processor offrono comandi per "copiare e incollare" e "tagliare e incollare" parti di testo. Anche l'editor `vi` dispone di queste funzioni. Nel caso di `vi`, l'operazione "copia e incolla" viene eseguita con i comandi *yank* e *put*, mentre l'equivalente di "taglia e incolla" si ottiene con i comandi *delete* e *put*.

Per copiare o spostare piccoli blocchi di testo sono disponibili varie combinazioni dei comandi `yank`, `delete` e `put`.

Copia di righe

Per copiare una riga sono richiesti due comandi: `yy` o `Y` ("yank") e `p` ("put below", inserisci sotto) o `P` ("put above", inserisci sopra). Si osservi che `Y` produce lo stesso effetto di `yy`.

Per copiare una riga, posizionare il cursore in qualsiasi punto della riga e digitare `yy`. Portare ora il cursore sulla riga sopra il punto in cui si desidera inserire la riga copiata e digitare `p`. Una copia della riga copiata verrà inserita *al di sotto* del cursore.

Per inserire la riga copiata *al di sopra* del cursore, digitare `P`.

Il comando `yy` può essere utilizzato anche con un numero: per copiare 11 righe, ad esempio, è possibile digitare `11yy`. `vi` copierà undici righe a partire da quella corrente, e visualizzerà al fondo dello schermo il messaggio: `11 righe memorizzate`.

I comandi `P` o `p` possono anche essere usati subito dopo uno dei comandi di cancellazione descritti in precedenza. In questo caso, il testo cancellato verrà inserito rispettivamente al di sopra o al di sotto del cursore.

Attenzione – Tra i comandi `yank`, `delete` e `put` usare solo comandi di spostamento del cursore. Se prima di inserire con `put` un testo tagliato o copiato si cancella o si copia un altro blocco di testo, la porzione tagliata o copiata in precedenza andrà perduta.

Spostamento di righe

Per lo spostamento di una o più righe di testo sono richiesti due comandi: `dd` ("delete") e `p` oppure `P`.

Per spostare una riga, posizionare il cursore in qualsiasi punto della riga e digitare `dd`. Per spostare più righe, far precedere il comando dal numero di righe su cui si desidera eseguire l'operazione. Per cancellare ad esempio 5 righe, digitare `5dd`.

Portare quindi il cursore sulla riga al di sopra del punto in cui si desidera reinserire la riga cancellata e digitare `p`. Il testo viene inserito in una nuova riga al di sotto del cursore.

Per inserire la riga cancellata al di sopra del cursore, digitare `P`.

Uso di nomi per i buffer

Per inserire ripetutamente un gruppo di righe in più punti di un documento, può essere comodo copiare le righe di interesse in un buffer e assegnare a questo buffer un nome. Per eseguire questa operazione, far precedere il comando da eseguire da una virgoletta doppia (`"`) e dal nome del buffer. Ad esempio, per copiare quattro righe in un buffer di nome *a*, digitare `"a4yy`. È possibile usare diversi buffer per uno stesso documento. Ad esempio, si può cancellare un testo da una data posizione ed aggiungerlo in diversi altri punti. Per cancellare 12 righe e inserirle in un buffer di nome *b*, digitare `"b12dd`.

Per inserire nel documento il testo contenuto nel buffer, far precedere il comando `p` o `P` da `"`*n*, dove *n* è il nome del buffer. Ad esempio, per inserire nel documento le righe salvate nel buffer *b*, digitare `"bP`.

Il contenuto di un buffer può essere sostituito con nuove righe di testo. I buffer creati durante una sessione restano in memoria fino all'uscita da `vi`.

L'uso dei buffer denominati permette di cancellare e copiare diversi blocchi di testo separatamente, senza sovrascrivere ad ogni operazione il contenuto del buffer principale.

Uso di contatori per la ripetizione dei comandi

Molti comandi di `vi` possono essere preceduti da un fattore di ripetizione, detto *contatore* — un numero che precede il comando e indica per quante volte l'operazione deve essere ripetuta.

Molti dei comandi descritti nei paragrafi precedenti possono essere utilizzati con un contatore. Ad esempio, `3dd` indica che il comando di cancellazione di una riga deve essere ripetuto per tre volte; il risultato sarà quindi la cancellazione di tre righe. Analogamente, `2dw` cancellerà due parole, mentre `4x` cancellerà quattro caratteri o spazi. I contatori possono essere abbinati anche ai comandi per lo spostamento del cursore, ad esempio `3w` e `2Ctrl-F`. Tutte queste funzioni diventeranno più chiare con l'uso pratico dei comandi di `vi`.

Nel paragrafo “Sommario dei comandi principali di vi” alla fine di questo capitolo, i comandi che accettano l’uso di un contatore sono indicati con “[numero]” prima del nome del comando.

Digitando un punto (`.`), `vi` ripete il comando di modifica precedente. Se ad esempio è stata cancellata una riga con `dd`, è possibile portare il cursore su un’altra riga e cancellarla digitando semplicemente un punto.

Uso dei comandi di `ex`

I comandi di `ex` possono essere più comodi e precisi di `yank`, `delete` e `put` quando le operazioni riguardano blocchi di testo relativamente estesi. Anziché contare le righe sullo schermo e quindi spostarsi nel nuovo punto di inserimento, si indica a `vi` un ambito di righe da spostare o da copiare e quindi si specifica la riga che precede il punto di inserimento desiderato.

Abilitazione e disabilitazione dei numeri di riga

Per *abilitare* i numeri di riga, digitare `:set nu` e premere Return.

I numeri di riga compaiono sul lato sinistro dello schermo. Non compaiono tuttavia quando il file viene stampato.

```
 1 Ascoltami, i poeti laureati
 2 si muovono soltanto tra le piante
 3 dai nomi poco usati: bossi ligustri o acanti.
 4
 5 Io, per me, amo le strade che riescono agli erbosi
 6 fossi dove in pozzanghere
 7 mezzo seccate agguantano i ragazzi
 8 qualche sparuta anguilla:
 9 le viuzze che seguono i ciglioni,
10 discendono tra i ciuffi delle canne
11 e mettono negli orti, tra gli alberi dei limoni.
~
~
~
~
:set nu
```

Per *disabilitare* i numeri di riga, digitare `:set nonu` e premere Return.

Copia di righe

La sintassi base del comando di `ex` per la copia di un testo è la seguente:

```
:numeroriga,numeroriga co numeroriga
```

I primi due numeri (separati da una virgola) specificano l'ambito di righe da copiare. Il terzo numero indica la riga *prima* del punto di inserimento.

Ad esempio, per copiare le righe da 1 a 5 di `prova` e inserire la copia dopo la riga 12, digitare:

```
:1,5 co 12
```

e premere Return.

Per specificare gruppi di righe, usare le seguenti abbreviazioni:

- Un punto (`.`) per indicare "dalla riga corrente".
- Il simbolo del dollaro ($) per indicare "fino alla fine del file".

Per copiare perciò "dalla riga corrente fino alla riga 5" e inserire il blocco copiato dopo la riga 12, il comando da digitare sarà il seguente:

```
:.,5 co 12
```

Per copiare "dalla riga 6 alla fine del file" e inserire il blocco copiato dopo la riga 2, il comando da digitare sarà:

```
:6,$ co 2
```

Spostamento di righe

La sintassi base del comando di spostamento di `ex` è simile a quella del comando di copia descritto in precedenza:

```
:numeroriga,numeroriga m numeroriga
```

Per specificare i gruppi di righe e i punti di inserimento si utilizzano le stesse convenzioni del comando di copia, compreso l'uso delle abbreviazioni `.` e `$`. L'unica differenza consiste nel fatto che il comando di spostamento cancella il blocco dalla posizione originaria e lo reinserisce nel punto designato.

Ad esempio, per spostare le righe 1-5 alla riga dopo la 12, digitare:

```
:1,5 m 12
```

e premere Return.

Cancellazione di righe

Per cancellare un gruppo di righe, la sintassi del comando è la seguente:

```
:numeroriga,numeroriga d
```

Ad esempio, per cancellare le righe 1-5, digitare:

```
:1,5 d
```

Ricerca e sostituzione con `vi`

`vi` offre diversi metodi per ricercare un punto specifico di un file usando una determinata stringa di caratteri. Dispone inoltre di una potente funzione di sostituzione globale.

Ricerca di una stringa di caratteri

Una *stringa di caratteri* è una qualsiasi sequenza di uno o più caratteri. Può comprendere lettere, numeri, segni di punteggiatura, caratteri speciali, spazi, tabulazioni o ritorni a capo. Una stringa può essere una parola intera o una parte di una parola.

Per ricercare una stringa di caratteri, digitare una barra (`/`) seguita dalla stringa desiderata e premere Return. `vi` posizionerà il cursore sulla ricorrenza successiva della stringa specificata. Ad esempio, per ricercare la stringa “meta”, digitare /meta e quindi Return.

Digitare `n` per spostarsi alla ricorrenza *successiva* della stringa, o `N` per tornare alla ricorrenza *precedente*.

Per eseguire una ricerca all’indietro si utilizza il carattere `?` al posto di `/`. Gli effetti di `n` e `N` per lo spostamento in avanti e all’indietro vengono invertiti.

I comandi di ricerca distinguono di norma tra maiuscole e minuscole: se la stringa impostata è “cina”, nel processo di ricerca non verrà trovata la parola “Cina”. Per fare in modo che `vi` ignori la differenza tra maiuscole e minuscole durante la ricerca, digitare `:set ic`. Per ripristinare l’impostazione di default, digitare `:set noic`.

Se `vi` trova la stringa ricercata, il cursore si ferma alla sua prima ricorrenza. Se la stringa non viene trovata, `vi` visualizza sull’ultima riga dello schermo il messaggio `Stringa non trovata`.

Alcuni caratteri speciali (`/ & ! . ^ * $` \ ?) hanno un significato particolare per il comando di ricerca, e per essere utilizzati all’interno di una stringa di ricerca devono essere preceduti dal carattere di escape (\). Ad esempio, per ricercare la stringa “qualcosa?” digitare `/qualcosa\?` e premere Return.

Questi caratteri speciali possono essere utilizzati come comandi per la funzione di ricerca; per essere interpretati in modo letterale nella stringa da ricercare devono essere preceduti da una barra rovesciata. Per utilizzare in senso letterale una barra rovesciata, digitare \\.

Funzioni avanzate di ricerca

Per rendere i processi di ricerca più precisi, è possibile inserire nella stringa da ricercare speciali indicatori per le seguenti caratteristiche:

- Inizio della riga
- Fine della riga
- Inizio della parola
- Fine della parola
- Caratteri speciali

Per ricercare una stringa collocata all’inizio di una riga, far precedere la stringa da un accento circonflesso (^). Ad esempio, per ricercare la prima riga che inizia con “Ricerca”, digitare:

```
/^Ricerca
```

Per ricercare una stringa collocata alla fine di una riga, digitare alla fine della stringa il simbolo del dollaro ($). Ad esempio, per ricercare la prima riga che termina con la stringa “ricerca.”, digitare:

```
/ricerca\.$
```

(Si osservi che il punto è preceduto da una barra rovesciata.)

Per ricercare una stringa posta all’inizio di una parola, digitare `\<` all’inizio della stringa; per ricercare una stringa posta alla fine di una parola, digitare `\>` alla fine della stringa. Per ricercare una parola intera, combinare le opzioni di inizio-parola e fine-parola nel modello di ricerca. Ad esempio, per ricercare la ricorrenza successiva della parola “ricerca” anziché della stringa “ricerca”, digitare:

```
/\<ricerca\>
```

Per ricercare un qualsiasi carattere in una data posizione, digitare un punto (`.`) nella posizione desiderata della stringa. Ad esempio, per ricercare la ricorrenza successiva di “formato” o “formati”, digitare:

```
/format.
```

In questo caso, poiché la ricerca si riferisce a una stringa e non a una parola, potranno essere trovati anche termini come “disinformato” o “informatica”.

Per ricercare caratteri alternativi in una stringa, indicare le alternative racchiuse tra parentesi quadre. Il modello di ricerca `/[md]`*stringa* troverà le stringhe che iniziano con m o con d. Specificando invece il modello di ricerca `/[d-m]`*stringa*, verranno trovate tutte le stringhe che iniziano con una qualsiasi lettera dalla d alla m.

Per cercare nessuna o più ricorrenze di un carattere, digitare un asterisco (*) dopo il carattere desiderato. Utilizzando in combinazione le parentesi quadre e l'asterisco è possibile ricercare specifici modelli alternativi. Ad esempio, per ricercare tutte le stringhe che iniziano con una lettera dalla a alla z e finiscono con "formato" *e* tutte le ricorrenze della stringa "formato", digitare:

```
/[a-z]*formato
```

Sostituzione di una stringa di caratteri

La procedura per la sostituzione di una stringa di testo è basata sulle procedure di ricerca descritte in precedenza. Tutti i caratteri speciali disponibili per la ricerca possono essere utilizzati anche per le operazioni di ricerca e sostituzione.

La sintassi base del comando è la seguente:

```
:g/stringa-di-ricerca/s//stringa-di-sostituzione/g
```

seguita da Return.

Per sostituire ad esempio tutte le ricorrenze della stringa "formato" con la stringa "dimensioni" digitare:

```
:g/formato/s//dimensioni/g
```

e premere Return.

Questo comando può essere modificato in modo che `vi` fermi la ricerca e chieda conferma della sostituzione ad ogni ricorrenza della stringa. Nel comando dell'esempio seguente, le opzioni `gc` specificano a `vi` di fermarsi ad ogni ricorrenza di "formato" e di chiedere conferma della sostituzione. Rispondere con `y` per sì o con `n` per no.

```
:g/formato/s//dimensioni/gc
```

Nota – Per annullare una procedura di ricerca e sostituzione con richiesta di conferma, premere Ctrl-C.

Spostamento a una riga specifica

Per spostarsi sull'ultima riga del file corrente, digitare `G`. Per tornare alla prima riga, digitare `1G`.

Per spostarsi su una qualsiasi altra riga specifica, digitare il numero della riga seguito da `G`.

Se ad esempio il file `prova` viene chiuso quando ci si trova alla riga 51, all'apertura successiva del file è possibile tornare direttamente a quella riga digitando `51G`.

Inserimento di un file in un altro

`vi` permette di inserire un intero file all'interno del file corrente. La sintassi generale del comando è la seguente:

```
:numeroriga r nomefile
```

Se non viene specificato il numero di riga, `vi` inserisce il file nella posizione corrente del cursore.

Per inserire ad esempio il file `montale` alla riga 84 del file `prova`, il comando da digitare è il seguente:

```
:84 r montale
```

Oppure, è possibile posizionare il cursore alla riga 84 e digitare:

```
:r montale
```

Apertura contemporanea di più file

vi permette di tenere aperti più file contemporaneamente. Ad esempio, per aprire e modificare il file montale mentre è già aperto il file prova, procedere come segue:

1. **Salvare il lavoro corrente in** prova**. Digitare** :w **e premere Return.**
2. **Aprire** montale **digitando** :n montale**, quindi premere Return.**
3. **Apportare le modifiche desiderate a** montale **e salvare il file.**
4. **Dopo avere modificato e salvato il file** montale**, sono disponibili tre alternative:**
   - Uscire da vi. Digitare :q e premere Return.
   - Tornare a prova. Digitare :n # e premere Return.
   - Lavorare alternatamente sui due file utilizzando il comando :n #.

Apertura di più file

Per aprire una serie di file, digitare vi seguito dai nomi dei file desiderati al prompt di sistema:

```
$ vi prova montale
```

I file vengono aperti nell'ordine in cui sono stati elencati. Nel caso dell'esempio verrà visualizzato per primo il file prova. Al termine del lavoro su prova, digitare :n per passare al file successivo, montale. Per passare al file successivo senza salvare le modifiche al file corrente, digitare :n! anziché :n.

Se i file da aprire hanno nomi simili (ad esempio, test1, test2, test3), è possibile specificarne i nomi utilizzando i caratteri speciali:

```
$ vi test*
```

I file verranno visualizzati in ordine alfabetico.

Copia di righe tra più file

Per copiare un gruppo di righe da un file in un altro, procedere come segue:

1. **Aprire il primo file.**
2. **Salvare le righe desiderate in un buffer usando il comando `yank`. Ad esempio, per salvare 10 righe nel buffer *a*, digitare `a10Y`.**
3. **Senza uscire da `vi`, passare al file successivo (`montale` in questo esempio):**

```
:n montale
```

4. **Inserire le righe copiate dal primo file con il comando `put`. Ad esempio, per inserire i contenuto del buffer *a* al di sotto della posizione corrente del cursore, digitare `ap`.**

Si ricordi che il contenuto dei buffer denominati vengono cancellati all'uscita da `vi`. È consigliabile perciò terminare le operazioni che comportano l'uso dei buffer prima di digitare il comando di uscita (`:q`).

Impostazione dei parametri di `vi`

`vi` dispone di molte variabili che agiscono sul funzionamento e sul tipo di visualizzazione dell'editor. Per visualizzare l'elenco delle variabili (con le impostazioni correnti) durante l'utilizzo di `vi`, digitare:

```
:set all
```

seguito da Return.

6

Recupero di un file dopo un crash del sistema

Un crash del sistema può causare la perdita del contenuto dei buffer. Spesso, tuttavia, è possibile recuperare la maggior parte del lavoro riavviando `vi` con il comando:

```
vi -r nomefile
```

dove *nomefile* è il file che era aperto al momento del crash. Dopo il riavvio, il sistema visualizza un messaggio indicante la presenza di un file di recupero.

Sommario dei comandi principali di `vi`

La tabella seguente riassume i comandi principali di `vi`.

Tabella 6-1 Comandi principali di `vi`

Comando	**Effetto**
Avvio di `vi`	
`vi` *nomefile*	Apre o crea un file
`vi`	Apre un nuovo file senza nome
`vi -r` *nomefile*	Recupera un file dopo un crash
`view` *nomefile*	Apre un file in modalità di sola lettura
Spostamento del cursore	
h	Di un carattere a sinistra
j	Alla riga successiva
k	Alla riga precedente
l	Di un carattere a destra
w	Di una parola a destra
W	Di una parola a destra (ignorando la punteggiatura)
b	Di una parola a sinistra
B	Di una parola a sinistra (ignorando la punteggiatura)
e	Alla fine della parola corrente

Tabella 6-1 Comandi principali di `vi` (Continua)

Comando	Effetto
Return	Alla riga successiva
Back Space	Di un carattere a sinistra
Barra spaziatrice	Di un carattere a destra
H	All'inizio dello schermo
M	A metà dello schermo
L	Al fondo dello schermo
Ctrl-F	Di una videata in avanti
Ctrl-D	Di mezza videata in avanti
Ctrl-B	Di una videata all'indietro
Ctrl-U	Di mezza videata all'indietro
Inserimento di caratteri e righe	
a	Inserisce caratteri a destra del cursore
A	Inserisce caratteri alla fine della riga
i	Inserisce caratteri a sinistra del cursore
I	Inserisce caratteri all'inizio della riga
o	Inserisce una riga sotto il cursore
O	Inserisce una riga sopra il cursore
Modifica del testo	
cw	Cambia la parola (o una sua parte) a destra del cursore
c	Cambia l'intera riga
C	Cambia dal cursore a fine riga
s	Sostituisce i caratteri dal cursore in avanti
r	Sostituisce il carattere sotto il cursore
`r` Return	Spezza la riga
J	Unisce la riga corrente a quella successiva
xp	Inverte il carattere corrente con quello alla sua destra
~	Cambia da maiuscolo a minuscolo (e viceversa)

Tabella 6-1 Comandi principali di `vi` (Continua)

Comando	Effetto
u	Annulla il comando precedente
U	Annulla tutte le modifiche alla riga corrente
:u	Annulla l'ultimo comando di `ex`
Cancellazione del testo	
x	Cancella il carattere sotto il cursore
X	Cancella il carattere a sinistra del cursore
dw	Cancella la parola (o la sua parte a destra del cursore)
dd	Cancella la riga su cui si trova il cursore
D	Cancella la parte della riga a destra del cursore
dG	Cancella fino alla fine del file
d1G	Cancella dall'inizio del file al cursore
:5,10 d	Cancella le righe 5-10
Copia e spostamento del testo	
yy	Copia una riga
Y	Copia una riga
p	Inserisce la riga copiata o cancellata sotto quella corrente
P	Inserisce la riga copiata o cancellata sopra quella corrente
:1,2 co 3	Copia le righe 1-2 e le inserisce dopo la riga 3
:4,5 m 6	Sposta le righe 4-5 dopo la riga 6
Impostazione dei numeri di riga	
:set nu	Mostra i numeri di riga
:set nonu	Nasconde i numeri di riga
Impostazione del riconoscimento di maiuscole e minuscole	
:set ic	Ignora maiuscole e minuscole nella ricerca
:set noic	Distingue tra maiuscole e minuscole nella ricerca
Ricerca di una riga	
G	Va all'ultima riga del file

Tabella 6-1 Comandi principali di `vi` (Continua)

Comando	Effetto
1G	Va alla prima riga del file
21G	Va alla riga 21
Ricerca e sostituzione	
`/`*stringa*	Ricerca *stringa*
`?`*stringa*	Ricerca *stringa* all'indietro
n	Ricerca la ricorrenza successiva di *stringa*
N	Ricerca la ricorrenza precedente di *stringa*
:g / *stringa-ricerca* / s / / *stringa-sostituzione* / g	Ricerca e sostituisce
Controllo dello schermo	
Ctrl-L	Ridisegna lo schermo
Inserimento di un file in un altro file	
`:r` *nomefile*	Inserisce un file dopo il cursore
`:34 r` *nomefile*	Inserisce un file dopo la riga 34
Salvataggio e uscita	
:w	Salva le modifiche (memorizza il contenuto del buffer)
:w *nomefile*	Scrive il contenuto del buffer nel file specificato
:wq	Salva le modifiche ed esce da vi
ZZ	Salva le modifiche ed esce da vi
:q!	Esce senza salvare le modifiche

La posta elettronica

7

SunOS comprende un programma di nome `mailx` per l'invio e il ricevimento di messaggi di posta elettronica (*email*). `mailx` dispone di funzioni per la lettura, la scrittura, l'invio, il ricevimento, il salvataggio e la cancellazione dei messaggi. Poiché non richiede un ambiente a finestre, il programma `mailx` può essere utilizzato su qualsiasi terminale. Può inoltre rappresentare una pratica alternativa, soprattutto per l'invio di messaggi brevi, al programma di posta elettronica a finestre. `mailx` supporta anche l'uso di alias postali.

Nota – Se si sta lavorando in ambiente OpenWindows, uscire da Posta Elettronica prima di eseguire gli esempi descritti in questo capitolo. L'attivazione di due processi di posta elettronica potrebbe infatti generare messaggi di errore e di avvertimento. L'invio di messaggi da una Finestra di comando o da una Shell in genere non crea alcun problema, mentre il salvataggio e la cancellazione dei messaggi hanno effetto sul "contenitore di default" di Posta Elettronica, causando confusione nell'applicazione.

Concetti fondamentali di `mailx`

In questo paragrafo sono descritte le funzioni e i comandi essenziali di `mailx`. Nei paragrafi successivi verranno descritte funzioni più complesse, ideate per un utilizzo più avanzato del programma.

Nel programma `mailx`, l'indirizzo per l'invio di un messaggio è formato dal nome di login e dal nome del sistema del destinatario. Se il destinatario utilizza lo stesso sistema del mittente è sufficiente specificare il nome di login.

Ogni utente ha una propria *casella postale* per il ricevimento dei messaggi. La casella postale si trova di solito nella directory `/var/mail/`*nomeutente*, dove *nomeutente* è il nome di login dell'utente.

`mailx` informa l'utente dell'arrivo di nuovi messaggi e li colloca nella casella postale. Quando i messaggi sono stati letti, `mailx` li dispone automaticamente in un file d'archivio di nome `mbox`, situato nella directory home dell'utente.

Avvio di `mailx`

Per avviare `mailx`, digitare il seguente comando al prompt di sistema e premere Return:

```
$ mailx
```

Se non vi sono nuovi messaggi in attesa, il terminale visualizza il messaggio:

```
Nessun messaggio per nomeutente
$
```

dove *nomeutente* è il nome di login dell'utente.

Invio di un messaggio di prova a se stessi

Per avere una prima idea del funzionamento di `mailx`, si provi ad inviare un messaggio a se stessi. Al prompt di sistema, digitare il comando `mailx` seguito dal proprio indirizzo (nome di login e nome del sistema). Ad esempio, se il nome di login fosse `lorenzo` e il nome del sistema fosse `firenze`, l'indirizzo completo dovrebbe essere scritto nella forma `lorenzo@firenze`. (Il simbolo @ viene usato con il significato di "a".) In una rete locale dovrebbe essere sufficiente il nome di login — consultare per sicurezza l'amministratore del sistema.

```
$ mailx lorenzo@firenze
```

Il programma risponderà con il prompt `Subject`:

```
$ mailx lorenzo@firenze
Subject:
```

Se si desidera, inserire una breve descrizione del messaggio e premere Return. Digitare ora il testo vero e proprio del messaggio; scrivere alcune righe brevi e premere Return alla fine di ogni riga. (Gli errori di battitura possono essere corretti tornando indietro con il tasto Back Space e digitando nuovamente il testo, ma solo *prima* di premere il tasto Return.)

L'esempio che segue mostra un possibile messaggio di prova (gli spazi tra le righe si ottengono premendo due volte il tasto Return):

```
$ mailx lorenzo@firenze
Subject: cogli l'attimo

Caro Lorenzo,

Quant'è bella giovinezza,
che si fugge tuttavia!
Chi vuol esser lieto, sia:
di doman non c'è certezza.


A presto,

Il Magnifico
```

Per inviare il messaggio, premere Return per completare l'ultima riga e quindi Ctrl-D. Dopo l'invio del messaggio, sullo schermo ricompare il prompt di sistema.

Lettura del messaggio di prova

Per leggere il messaggio di prova, eseguire nuovamente il comando `mailx`. Comparirà una videata simile alla seguente:

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Digitare ? per
informazioni.
“/var/mail/lorenzo”: 2 messaggi 1 nuovi
 U  2 max@meteora      Fri Feb 14 12:01   14/318 aggiornamenti
>N  1 lorenzo@firenze  Mon Feb 17 08:12   21/453 cogli l’attimo
&
```

La prima riga identifica la versione di `mail`; la seconda riga indica la casella postale, di solito collocata in `/var/mail/`*nomeutente*, in cui viene depositata la posta in arrivo. La quarta riga dell’esempio è l’intestazione del messaggio di prova. La “N” all’inizio della riga indica che si tratta di un messaggio “nuovo”. La “U” che precede la terza riga dell’esempio (da “unread”, non letto) indica un messaggio che non è stato letto in una sessione precedente di `mailx`. (Le informazioni di questa videata saranno trattate in maggiore dettaglio nel paragrafo “Lettura dei messaggi” di questo capitolo.)

Ad ogni messaggio ricevuto viene assegnato un numero: il messaggio di Lorenzo a se stesso compare nell’esempio con il numero `1`.

Per leggere un messaggio, digitare il suo numero al prompt di `mailx`, rappresentato da una “e commerciale” (&):

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Digitare ? per
informazioni.
“/var/mail/lorenzo”: 1 messaggi 1 nuovi
>N  1 lorenzo@firenze   Fri Jul 14 12:01 21/453 cogli l’attimo
& 1

To: lorenzo@firenze
From: lorenzo@firenze
Subject: cogli l’attimo

Caro Lorenzo,

Quant’è bella giovinezza,
che si fugge tuttavia!
Chi vuol esser lieto, sia:
di doman non c’è certezza.


A presto,

Il Magnifico

&
```

Uscita da `mailx`

Per uscire da `mailx` sono disponibili due comandi: `q` o `x`.

Digitando `q` al prompt di `mailx` e premendo Return,

```
& q
```

compare un messaggio simile al seguente:

`1 messaggio salvato in` *directory_home*`/mbox`.

dove *directory_home* è il percorso della directory home dell’utente.

Se si utilizza il comando di uscita `q` dopo avere letto i messaggi ricevuti, `mailx` sposta i messaggi dalla casella postale al file `mbox` nella directory home dell'utente. `mailx` salva anche eventuali modifiche o cancellazioni dei messaggi.

Digitando invece il comando di uscita `x` al prompt di `mailx` e premendo Return,

```
& x
```

`mailx` *non* salva le eventuali modifiche o cancellazioni, né sposta i messaggi già letti nel file `mbox`.

Lettura dei messaggi

Se sono arrivati nuovi messaggi, `mailx` visualizza ad ogni login il messaggio

`You have mail`

o

You have new mail

Per leggere i messaggi, digitare `mailx` al prompt di sistema e premere Return. Se la casella postale è vuota compare il messaggio:

`Nessun messaggio per` *nomeutente*

Diversamente, compare una lista simile alla seguente:

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Digitare ? per
informazioni.
"/var/mail/lorenzo": 4 messaggi 1 nuovi 2 non letti
    1 lorenzo@firenze   Fri Feb 14 12:01 21/453 cogli l'attimo
 U  2 boccia@biliardo   Fri Feb 14 18:31 19/353 strategia
 U  3 fieno@fattoria    Sat Feb 15 10:22 24/557 previsioni per
>N  4 hoover@woofer     Sun Feb 16 23:59 14/280 attenzione agli
&
```

Il programma mailx visualizza alcune informazioni su se stesso (numero di versione e data) e le istruzioni per la richiesta della guida (Digitare ? per informazioni).

Nella riga successiva, mailx specifica la posizione della casella postale, il numero dei messaggi ricevuti e il loro stato.

mailx presenta quindi un elenco numerato dei messaggi presenti nella casella postale. Da sinistra a destra, le colonne di ogni riga indicano le seguenti informazioni:

- *Stato*: specifica se il messaggio è nuovo (N), non letto (U) o già letto (nessun simbolo). Il simbolo “>” all’inizio di una riga indica il messaggio corrente. I messaggi cancellati sono contrassegnati con un asterisco (*).
- *Numero*: indica l’ordine di ricevimento del messaggio.
- *Mittente*: indica il nome dell’utente (di norma nella forma nome@macchina) che ha inviato il messaggio.
- *Ora*: indica la data e l’ora di spedizione del messaggio.
- *Dimensioni*: indica il numero di righe / caratteri del messaggio.
- *Oggetto*: indica l’oggetto del messaggio specificato dal mittente.

Se la casella postale contiene molti messaggi, la lista visualizzata potrebbe non essere completa. In questo caso, digitare:

- z – per visualizzare la videata di intestazioni successiva
- h- – per visualizzare la videata di intestazioni precedente
- h – per rivisualizzare da qualsiasi punto la lista delle intestazioni

Per visualizzare il messaggio corrente (contrassegnato nella lista con >), premere Return. Premendo nuovamente Return verrà visualizzato il messaggio successivo. Per leggere un determinato messaggio, digitarne il numero e premere Return.

Cancellazione (e recupero) dei messaggi

Dopo avere letto un messaggio, si può decidere di cancellarlo anziché memorizzarlo nel file mbox (il salvataggio viene eseguito per default all’uscita da mailx).

Per cancellare l'ultimo messaggio letto, digitare `d` al prompt di `mailx`. Per cancellare un messaggio specifico dalla casella postale, usare il comando:

`d` *numero*

Per cancellare ad esempio il secondo messaggio, digitare il comando seguente:

```
& d 2
```

È anche possibile cancellare diversi messaggi con un'unica operazione. Per cancellare i messaggi 1 *e* 3, digitare il comando:

```
& d 1 3
```

Per cancellare un gruppo di messaggi consecutivi, ad esempio *dall'*1 *al* 3, digitare il comando:

```
& d 1-3
```

Prima di uscire da `mail` è ancora possibile *recuperare* i messaggi cancellati dalla casella postale. Digitare il comando:

`u` *numero*

seguito da Return. Per recuperare ad esempio il secondo messaggio, digitare il comando:

```
& u 2
```

Per annullare l'ultima cancellazione è possibile digitare `u` al prompt di `mailx` subito dopo la cancellazione. Ad esempio, se l'ultimo comando di cancellazione è stato `d 2-5`, il comando `u` recupererà i messaggi `2`, `3`, `4` e `5`.

Si ricordi che le cancellazioni diventano permanenti se si esce da `mailx` con il comando `q`; in altre parole, dopo l'uscita dal programma con `q` i messaggi non possono più essere recuperati. È però possibile uscire da `mailx` con il comando

x lasciando la casella postale intatta — come accennato in precedenza, uscendo con x i messaggi letti resteranno contrassegnati con una U, i messaggi cancellati verranno ripristinati, ecc.

Stampa dei messaggi

Per stampare un messaggio è possibile inviarlo a un comando di stampa utilizzando il simbolo di pipe, digitando:

| *numero* lp

al prompt di mailx. (Il simbolo | viene chiamato *pipe*.) Ad esempio, per stampare una copia del messaggio 2, digitare:

```
& |2 lp
```

e premere Return. Se non viene specificato il numero del messaggio, mailx invia alla stampante il messaggio corrente. Per maggiori informazioni sull'uso del simbolo di pipe, vedere “Redirezionamento e piping dell'output di un comando” al Capitolo 2, “Comandi principali di SunOS”.

Invio dei messaggi

Per inviare un messaggio con mailx occorre conoscere il nome di login del destinatario. Se il destinatario opera su un sistema diverso dal proprio, occorre conoscere anche il nome del suo sistema. Per ottenere queste informazioni è possibile usare i comandi who, finger o rusers.

Il comando who elenca tutti gli utenti correntemente connessi al proprio file server. La lista visualizzata indica, per ogni utente, il nome di login, il tipo di terminale e la data e l'ora di login. Ad esempio:

```
$ who
    bruno      tty15        feb 20 10:22
    mara       tty04        feb 20 10:37
    paola      tty07        feb 20 11:49
    laura      tty06        feb 20 12:02
```

Il comando `finger` visualizza le stesse informazioni ottenute con `who`, ma in forma più dettagliata. Le informazioni visualizzate dipendono dalla configurazione impostata dall'amministratore del sistema. Un esempio di output è il seguente:

```
$ finger
    Login     Name          TTY      Idle     When
    bruno     Bruno Rossi   tty15     43      Thu 10:22
    mara      Mara Bianchi  tty04             Thu 10:37
    paola     Paola Verdi   tty07     12      Thu 11:49
    laura     Laura Polli   tty06     22      Thu 12:02
```

Il comando `rusers` visualizza informazioni sugli utenti correntemente connessi alla rete locale. Per istruzioni sull'uso del comando `rusers`, vedere il Capitolo 9, "Operazioni in rete".

Dopo avere ottenuto le informazioni necessarie sull'utente, procedere come segue per inviare il messaggio.

1. **Digitare il comando `mailx` seguito dall'indirizzo dell'utente:**

```
$ mailx nomeutente@nomesistema
```

dove *utente* è il nome di login del destinatario e *sistema* è il nome del suo sistema.

- Se `mailx` è già stato avviato, è sufficiente digitare una `m` seguita dal nome di login e dal nome di sistema del destinatario:

```
& m nomeutente@nomesistema
```

- Per inviare lo stesso messaggio a più destinatari, separare gli indirizzi con spazi o virgole; ad esempio:

```
$ mailx boccia@biliardo paola@fattoria lorenzo@firenze
```

oppure

```
$ mailx boccia@biliardo,paola@fattoria,lorenzo@firenze
```

2. **Premendo Return,** `mailx` **richiede l'oggetto del messaggio. Inserire una breve descrizione del contenuto del messaggio e premere nuovamente Return.**

3. **Digitare il contenuto del messaggio. Per creare una riga nuova, premere Return.**
   Si noti che anche se una frase si sviluppa su più righe, il programma la considera come un'unica riga finché non viene premuto il tasto Return.

Nota – Ogni riga di testo all'interno del messaggio può contenere fino a 256 caratteri. Se questo limite viene superato lo schermo si blocca. In questo caso, premere Ctrl-C per abbandonare il messaggio.

4. **Dopo avere scritto il messaggio, premere Return per spostare il cursore su una nuova riga. Premere quindi Ctrl-D per inviare il messaggio.**

Errori di invio

Se l'indirizzo specificato per l'invio del messaggio non è corretto, il sistema risponde con il messaggio

nomeutente@nomesistema`...User unknown`

e il messaggio ritorna nella casella postale del mittente. All'avvio successivo di `mailx`, l'intestazione segnalerà la presenza di un messaggio restituito al mittente:

```
N 1 Mailer-Daemon Fri Jan 3 11:13 8/49 Returned mail: User unknown
```

Quando il messaggio non può essere consegnato, il file corrispondente viene copiato in un file della directory home con il nome `dead.letter`.

Annullamento di un messaggio

Per cancellare un messaggio *prima* di inviarlo, premere due volte Ctrl-C.

Copie per conoscenza e copie silenti dei messaggi

Prima di inviare un messaggio, è possibile impostare l'invio di "copie per conoscenza" ad utenti diversi dai destinatari principali. È anche possibile inviare "copie silenti". Chi riceve il messaggio può conoscere i destinatari delle copie per conoscenza ma non quelli delle copie silenti.

Può essere buona norma includere anche il proprio indirizzo tra i destinatari delle copie per conoscenza o delle copie silenti, in modo da conservare sempre una copia dei messaggi spediti.

Sono disponibili tre metodi per inviare copie per conoscenza di un messaggio:

- Aprire il file `.mailrc` (nella directory home) con un editor di testo e inserire la riga seguente:

```
set askcc
```

  Con questa impostazione, `mailx` visualizza il prompt per le copie per conoscenza (`Cc:`) dopo il prompt per l'oggetto. Inserire in questo campo gli indirizzi degli utenti desiderati separandoli con spazi vuoti.

- Dopo avere scritto il messaggio, ma prima di premere Ctrl-D, premere Return per creare una nuova riga e digitare il comando:

  `~c` *indirizzo/i*

  Se si utilizza questo metodo per inviare copie per conoscenza a più destinatari, separare gli indirizzi con spazi vuoti. Ad esempio:

```
~c biglia@biliardo paola@fattoria lorenzo@firenze
```

- Per creare una riga `Cc:` si può anche utilizzare il comando `~h`, che visualizza l'intera intestazione del messaggio. `~h` visualizza, una alla volta, le righe di prompt `To:`, `Subject:`, `Cc:` e `Bcc:` (copia silente). Da qui è possibile compilare le righe vuote e modificare quelle già compilate. Come per gli altri comandi tilde, `~h` deve essere digitato su una riga nuova.

Nota – I comandi `~c`, `~h` ed altri comandi tilde sono descritti nel paragrafo "Comandi tilde" in questo capitolo.

Inserimento di una copia di un messaggio o di un file

`mailx` permette di inserire nel messaggio che si sta scrivendo una copia di qualsiasi messaggio contenuto nella casella postale, oppure una copia di un qualsiasi file di testo.

Inserimento di un messaggio

Il comando per l'inserimento di un messaggio è il seguente:

`~m` *numero*

dove *numero* è il numero del messaggio da inserire. Ad esempio, per inviare un messaggio che comprenda una copia del messaggio numero 3 della propria casella postale, procedere come segue:

1. **Su una nuova riga, digitare il comando** `~m 3` **e premere Return.**
2. `mailx` **visualizza il messaggio** `Inserimento di: 3 (continuare)`
3. **Il messaggio** `3` **non compare sullo schermo del mittente, mentre sarà visibile nella copia ricevuta dal destinatario. Dopo** `(continuare)`**, è possibile continuare il messaggio o inviarlo nella forma corrente.**
4. **Per visualizzare il messaggio in forma completa, compreso il messaggio incorporato, usare il comando** `~p`**.**

Inserimento di un file

È anche possibile inserire in un messaggio una copia di un qualsiasi file di testo. Usare il comando seguente:

`~r` *nomefile*

mentre si scrive il messaggio. Ad esempio, per inserire il file `profilo` nel messaggio corrente, digitare:

```
~r profilo
```

Risposta a un messaggio

Per rispondere a un messaggio, digitare il comando

r *numero*

al prompt di `mailx`. (Se non viene specificato il numero del messaggio, `mailx` risponde al messaggio corrente.) Ad esempio, per rispondere al mittente del messaggio 2, digitare il comando:

```
& r 2
```

`mailx` inserisce automaticamente l'indirizzo del destinatario e una riga `Re: Subject:` analoga alla riga `Subject:` del messaggio originale. Inviare la risposta con la normale procedura di spedizione.

Una variante del comando di risposta è `R`, che invia il messaggio sia al mittente che a tutti i destinatari del messaggio originario. Usare questo comando solo quando è assolutamente necessario, per evitare di generare “posta inutile”.

Nota – Anche in una risposta è possibile inserire un altro messaggio usando la procedura descritta al paragrafo precedente. Per inserire una copia del messaggio al quale si sta rispondendo, digitare il comando `~m` senza specificare il numero del messaggio.

Salvataggio e richiamo dei messaggi

I messaggi inviati o ricevuti possono anche essere salvati e richiamati per utilizzi successivi. In `mailx` è possibile salvare i messaggi inserendoli in normali file di testo, oppure inserendoli in file speciali detti raccoglitori. Entrambi i metodi sono descritti nei paragrafi seguenti.

mailx distingue tra il *salvataggio* e la *copia* di un messaggio; il salvataggio elimina il messaggio dalla casella postale e lo colloca in un file o in un raccoglitore; la copia lascia il messaggio nella casella postale e ne colloca una copia in un file o in un raccoglitore.

Salvataggio e copia di un messaggio in un file

Per salvare un messaggio in un file, digitare al prompt di mailx il comando:

s *numero nomefile*

dove *numero* è il numero del messaggio da salvare e *nomefile* è il file in cui si desidera salvarlo. Ad esempio, per salvare il messaggio 3 in un file di nome ~/note/bilancio, digitare:

```
& s 3 ~/note/bilancio
```

(Si ricordi che, in un percorso, il simbolo ~ rappresenta la directory home.)

È anche possibile salvare diversi messaggi nello stesso file con un'unica operazione. Ad esempio, per salvare i messaggi 3, 5, 6, 7 e 8 in ~/note/bilancio, digitare:

```
& s 3 5-8 ~/note/bilancio
```

Se il file specificato non esiste, mailx ne crea uno nuovo. Se il file esiste, mailx aggiunge il nuovo messaggio alla fine del file.

Il salvataggio cancella il messaggio dalla casella postale; mailx visualizza un asterisco (*) vicino all'intestazione dei messaggi salvati.

Per salvare il messaggio in un file lasciandone una copia nella casella postale, usare il comando copy come segue:

```
& c 3 ~/note/bilancio
```

Salvataggio e copia dei messaggi in un raccoglitore

Per evitare di scrivere il percorso completo del file utilizzato per l'archiviazione, è possibile salvare o copiare i messaggi in un cosiddetto raccoglitore. I raccoglitori sono file speciali situati in un'apposita directory.

Utilizzando i raccoglitori, i messaggi salvati o copiati vengono riuniti automaticamente nella stessa directory, da dove è possibile richiamarli facilmente senza bisogno di specificare lunghi percorsi.

Creazione della directory dei raccoglitori

Per usare i raccoglitori è necessario creare un'apposita directory. Procedere come segue:

1. **Creare la directory con il comando `mkdir`.**
   Ad esempio, per creare una directory di nome `Messaggi`, digitare il comando:

```
$ mkdir Messaggi
```

2. **Aprire quindi con un editor il file `.mailrc` situato nella propria directory home (è il file che contiene le opzioni per `mailx`) e impostare il percorso della nuova directory.**
   Modificare la variabile `set folder` inserendo il percorso completo della directory appena creata. Ad esempio:

```
set folder=/home/corte/lorenzo/Messaggi
```

   In alternativa, se si sta utilizzando una C shell è possibile usare il simbolo ~ per specificare la directory home.

```
set folder=~/Messaggi
```

A questo punto la directory è impostata per ricevere i messaggi salvati nei raccoglitori. (La modifica al file `.mailrc` diventa effettiva all'avvio successivo di `mailx`.)

Designazione dei raccoglitori

Per salvare e copiare un messaggio in un raccoglitore o in un file si utilizzano gli stessi comandi, con la differenza che il nome del raccoglitore è preceduto da un segno (+) anziché da un percorso. Il segno + indica a `mailx` che il raccoglitore deve essere archiviato nella directory appositamente impostata (`Messaggi`).

Ad esempio, per salvare il messaggio 3 in un raccoglitore di nome `progetti`, digitare:

```
& s 3 +progetti
```

`mailx` interpreta questo comando come "salva il messaggio 3 in `~/Messaggi/progetti`". (Se il raccoglitore specificato non esiste, `mailx` lo crea al momento del salvataggio.)

Per *copiare* il messaggio nel raccoglitore, digitare:

```
& c 3 +progetti
```

Invio di un messaggio a un file o a un raccoglitore

È anche possibile inviare una copia dei messaggi spediti direttamente a uno dei propri file o raccoglitori. Per dirigere una copia in un raccoglitore, digitare il nome del raccoglitore nel campo `Cc:` o nel campo `Bcc:`. Per dirigere una copia del messaggio in un file, seguire la stessa procedura specificando il percorso completo del file.

Lettura dei messaggi salvati in file e raccoglitori

Per leggere un messaggio salvato in un file, usare il comando:

`mailx -f` *nomefile*

Ad esempio, per leggere il file `~/memo/spese`, digitare:

```
$ mailx -f ~/memo/spese
```

Per leggere un messaggio salvato in un raccoglitore, seguire la stessa procedura utilizzando il segno + al posto del percorso. Ad esempio, per leggere i messaggi memorizzati nel raccoglitore `progetti`, digitare:

```
$ mailx -f +progetti
```

Questo comando avvia `mailx` nel file o nel raccoglitore specificato. Verranno visualizzate solo le intestazioni dei messaggi presenti nel file o nel raccoglitore. Per selezionare uno dei messaggi, digitarne il numero al prompt di `mailx` e premere Return.

È anche possibile utilizzare i raccoglitori dopo avere avviato `mailx`. Per visualizzare l'elenco dei raccoglitori disponibili, digitare al prompt di `mailx` il comando:

```
& folders
```

Per passare dalla casella postale a un raccoglitore, digitare il comando:

```
& folder +nome_raccoglitore
```

Per tornare alla casella postale, digitare al prompt:

```
& %
```

Per tornare al raccoglitore precedente, digitare:

```
& #
```

Uso di `vi` *con* `mailx`

Per comporre un messaggio in `mailx` è anche possibile utilizzare l'editor di testo `vi`. In questo modo si possono correggere eventuali errori e aggiungere o cancellare parti di testo prima di inviare i messaggi. Per informazioni sull'uso di `vi`, vedere il Capitolo 6, "L'editor vi".

In mailx è possibile usare i comandi standard di vi per inserire, cancellare e modificare parti di testo.

Per scrivere un messaggio con vi:

1. **Digitare il comando mailx con un indirizzo al prompt di mailx (&) o al prompt di sistema ($).**
2. **Inserire l'oggetto del messaggio nel campo Subject: e premere Return.**
3. **Avviare vi digitando da una nuova riga il comando ~v. Appare la videata di vi contenente un file vuoto nella directory /tmp.**
4. **Usare i comandi di vi per inserire e modificare il testo del messaggio.**
5. **Al termine, uscire da vi con il comando :wq o ZZ.**

Dopo l'uscita da vi, mailx visualizza il messaggio (continuare):. A questo punto è possibile modificare ulteriormente il messaggio (al di fuori di vi) o inviarlo premendo Ctrl-D.

Alias postali

Un *alias postale* rappresenta un gruppo di utenti riuniti sotto uno stesso nome.

Gli alias postali vengono in genere utilizzati quando si inviano messaggi regolarmente a uno stesso gruppo di persone. Ad esempio, se capita spesso di inviare lo stesso messaggio a paola@fattoria, giorgio@sicilia e sandra@dakota, è possibile creare un alias postale di nome amici. Specificando amici come destinatario, il messaggio sarà ricevuto da tutti gli utenti riuniti sotto quell'alias.

Sono disponibili due file per l'impostazione degli alias postali:

- .mailrc e
- /etc/aliases

Gli alias impostati in .mailrc si comportano diversamente da quelli impostati in /etc/aliases. Le differenze sono riassunte nella Tabella 7-1 alla fine di questo paragrafo.

Impostazione di un alias postale in `.mailrc`

Per impostare un alias in `.mailrc` occorre ricordare quanto segue:

- Gli alias postali contenuti in `.mailrc` sono *privati*, possono cioè essere usati solo dall'utente proprietario. Ad esempio, se si definisce un alias di nome `amici` in `.mailrc` e un altro utente tenta di inviare un messaggio ad `amici`, il sistema genera il messaggio `unknown user` (utente sconosciuto).
- Dopo l'invio del messaggio, gli alias contenuti in `.mailrc` si espandono automaticamente mostrando gli utenti raggruppati sotto quel nome. Se ad esempio si invia un messaggio ad `amici`, il messaggio viene ricevuto come se il mittente avesse effettivamente scritto i nomi di tutti i destinatari. In altre parole, i destinatari non possono sapere che è stato utilizzato un alias postale.

Il file `.mailrc` si trova nella directory home e contiene una serie di parametri che controllano il comportamento di `mailx` e di Posta Elettronica.

Per inserire un alias postale in `.mailrc`, digitare:

```
$ vi ~/.mailrc
```

Nota – Il file `.mailrc` può essere aperto con qualsiasi editor di testo. L'esempio precedente illustra il comando da utilizzare per aprire il file con `vi`. Per informazioni sull'uso di `vi`, vedere il Capitolo 6, "L'editor vi".

Ogni alias postale deve essere contenuto in un'unica riga del file. Questa condizione non riguarda la forma di visualizzazione (l'alias può occupare più righe sullo schermo); significa piuttosto che l'alias non può contenere caratteri di ritorno a capo. Ogni alias deve contenere le informazioni seguenti, separate da spazi:

- La parola "alias"
- Il nome dell'alias (una sola parola)
- I destinatari (nome di login e nome del sistema) da includere nell'alias, separati da spazi.

L'esempio seguente mostra due alias postali. Il primo (amici) include tre persone, il secondo (softball) ne include otto. Si noti che in softball i nomi degli utenti occupano più di una riga sullo schermo, ma non vengono utilizzati caratteri di ritorno a capo.

```
alias amici paola@fattoria giorgio@sicilia sandra@dakota
alias softball ugo@woofer max@twister paola@fattoria gabri@zoo
lorenzo@firenze jj@walker sandra@dakota stefano@fiesta
```

Per inviare un messaggio alle persone riunite sotto un alias di .mailrc, è sufficiente utilizzare come destinatario l'alias corrispondente. *Non* occorre inserire il nome del proprio sistema. Ad esempio, inviando il messaggio seguente:

```
$ mail amici
Subject: Proposta mangereccia

Ciao a tutti. Cosa ne dite di pranzare insieme il prossimo
venerdì? C'è qualcuno libero?
```

i destinatari lo vedrebbero nel modo seguente (si osservi la riga To: espansa):

```
To: paola@fattoria giorgio@sicilia sandra@dakota
Subject: Proposta mangereccia

Ciao a tutti. Cosa ne dite di pranzare insieme il prossimo
venerdì? C'è qualcuno libero?
```

Impostazione di un alias postale in /etc/aliases

Per impostare un alias postale in /etc/aliases occorre ricordare quanto segue:

- Gli alias postali contenuti in /etc/aliases sono *pubblici*. Se ad esempio un utente imposta un alias di nome softball, chiunque potrà utilizzare softball@*nomesistema* come destinatario di un messaggio.

- Dopo l'invio del messaggio, gli alias di `/etc/aliases` *non* vengono espansi. Se ad esempio si invia un messaggio a `softball@`*sistema*, l'alias comparirà anche nell'intestazione della copia ricevuta dai destinatari. Questi conosceranno perciò il nome dell'alias utilizzato per l'invio del messaggio, ma non necessariamente i nomi degli altri utenti.

Gli alias creati in `/etc/aliases` hanno un formato leggermente diverso da quelli di `.mailrc`. Ogni alias di `/etc/aliases` deve essere composto come segue:

- Nome dell'alias seguito da due punti (`:`)
- Destinatari (nome di login e nome del sistema) separati da virgole. Si noti che l'alias *non* deve necessariamente essere scritto in un'unica riga.

Per modificare il file `/etc/aliases` occorre operare come utente root. Se l'accesso al sistema come utente root è protetto da una password, è necessario conoscere la password.

Per operare sul sistema come utente root, digitare quanto segue:

```
$ su
Password:
#
```

Si noti che il prompt di sistema utilizza un simbolo diverso per l'utente root.

L'esempio seguente mostra come aggiungere l'alias `softball@sport` al file di default `/etc/aliases`.

```
# vi /etc/aliases
##
#Aliases can have any mix of upper and lower case on the left-
#hand side,
#but the right-hand side should be proper case (usually lower)
#
#    >>>>>>>>>>The program “newaliases” will need to be run after
#     >> NOTE >>this file is updated for any changes to
#     >>>>>>>>>>show through to sendmail.
#
#@(#)aliases 1.10 89/01/20 SMI
##
```

```
# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system's
mail problems.
Postmaster: root

# Alias for mailer daemon; returned messages from our MAILER-
DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster

# Aliases to handle mail to programs or files, eg news or vacation
# decode: "|/usr/bin/uudecode"
nobody: /dev/null

# Sample aliases:
# Alias for distribution list, members specified here:
#staff:wnj,mosher,sam,ecc,mckusick,sklower,olson,rwh@ernie

# Alias for distribution list, members specified elsewhere:
#keyboards: :include:/usr/jfarrell/keyboards.list

# Alias for a person, so they can receive mail by several names:
#epa:eric

#######################
# Local aliases below #
#######################
softball@sport: ugo@woofer max@twister paola@fattoria gabri@zoo
lorenzo@firenze jj@walker sandra@dakota stefano@fiesta
:wq           (per uscire da vi e salvare il file /etc/aliases)
# exit        (per uscire dal modo root)
$
```

Il file `/etc/aliases` può essere aperto con qualsiasi editor di testo. L'esempio precedente illustra i comandi utilizzati per aprire il file con `vi`. Per informazioni sull'uso di `vi`, vedere il Capitolo 6, "L'editor vi".

Si noti che i cancelletti (#) che compaiono nel file `/etc/aliases` hanno lo scopo di *commentare* il testo e gli alias di esempio. Il simbolo del cancelletto impedisce al sistema di utilizzare queste informazioni come alias effettivi.

Non inserire un cancelletto davanti agli alias che si desidera aggiungere, a meno che non si voglia intenzionalmente disabilitarli.

Per inviare un messaggio alle persone riunite sotto un alias di `/etc/aliases`, indicare come destinatario il nome dell'alias e il nome del proprio sistema. Ad esempio, inviando il messaggio seguente:

```
$ mail softball@sport
Subject: Allenamento oggi

Appuntamento al campo numero 4 dopo il lavoro. Se tutto va bene
riusciamo ad allenarci per la partita di sabato! Cercate di
arrivare il più presto possibile.
```

i destinatari lo riceveranno in questa forma:

```
To: softball@sport
Subject: Allenamento oggi

Appuntamento al campo numero 4 dopo il lavoro. Se tutto va bene
riusciamo ad allenarci per la partita di sabato! Cercate di
arrivare il più presto possibile.
```

Si osservi che la riga `To:` *non* viene espansa.

Quando si utilizza un alias postale di questo tipo, occorre sempre specificare il nome del sistema nel quale l'alias è stato definito. Se ad esempio si imposta un alias di nome `piloti` sul sistema `circuito`, i messaggi dovranno essere inviati a `piloti@circuito`.

La Tabella 7-1 riassume le differenze tra gli alias creati in `.mailrc` e quelli creati in `/etc/aliases`.

Tabella 7-1 Confronto tra gli alias di `.mailrc` e quelli di `/etc/aliases`

	.mailrc	**/etc/aliases**
Modifica riservata a utente root	no	sì
Formato destinatario:	alias	*alias@nomesistema*
Nomi visibili dai destinatari	sì	no
Nomi separati da virgole	no	sì
Nomi su un'unica riga	sì	no
Alias accessibili ad altri utenti	no	sì

Per informazioni più dettagliate sugli alias postali, digitare `man aliases` o `man addresses` al prompt di sistema.

Comandi tilde

Durante la composizione di un messaggio è possibile utilizzare i cosiddetti comandi tilde per una serie di funzioni. I comandi tilde sono formati in genere da una tilde (~) seguita da una lettera. La tabella seguente descrive alcuni dei comandi tilde più comuni, alcuni dei quali sono già stati menzionati nel corso del capitolo.

Nota – Per inserire un vero carattere di tilde all'interno di un messaggio, digitare due caratteri tilde in sequenza; ne verrà visualizzato uno solo.

Tabella 7-2 Comandi tilde (`mailx`)

Comando	Funzione
~!*comando*	Permette l'esecuzione di un comando shell
~.	Simula Ctrl-D per marcare la fine del file
~?	Visualizza un elenco dei comandi tilde
`~b` *nomeutente*	Permette di aggiungere un nome utente ai destinatari delle copie silenti (Bcc:)
`~c` *nomeutente*	Permette di aggiungere un nome utente ai destinatari delle copie per conoscenza (Cc:)
~d	Inserisce il contenuto del file `dead.letter` nel messaggio corrente.
`~f` *numero*	Invia il messaggio specificato. Disponibile solo quando si invia un messaggio durante la lettura della posta.
~h	Visualizza i prompt per le righe di intestazione: Subject, To, Cc e Bcc.
`~m` *numero*	Inserisce il testo del messaggio specificato nel messaggio corrente. Disponibile solo quando si invia un messaggio durante la lettura della posta.
~p	Stampa il messaggio inserito.
~q	Simula un doppio Ctrl-C. Se il messaggio corrente non è vuoto, il contenuto viene salvato nel file `dead.letter`.
`~r` *nomefile*	Inserisce nel messaggio il testo del file specificato.

Tabella 7-2 Comandi tilde (`mailx`)

Comando	Funzione
`~s` *stringa*	Cambia la riga Subject: in *stringa*.
`~t` *nome*	Aggiunge i nomi specificati nel campo di intestazione To.
`~w` *nomefile*	Salva il messaggio corrente senza intestazione nel file specificato.
~x	Esce da `mailx`. Simile a ~q, ma il messaggio non viene salvato nel file `dead.letter`.

Visualizzazione della guida per gli altri comandi di `mailx`

`mailx` dispone di due comandi che visualizzano l'elenco dei comandi e delle funzioni disponibili. Se ci si trova in modalità comando, digitando `?` al prompt di `mailx` (&) compare un elenco dei comandi disponibili per quella modalità. Analogamente, se ci si trova in modalità inserimento (ad esempio durante la composizione di un messaggio) è possibile digitare il comando equivalente ~? per visualizzare un elenco dei comandi tilde (detti anche "escape tilde").

Le pagine del manuale online contengono informazioni approfondite su `mailx`, espresse in forma più tecnica. Per visualizzare tali informazioni, digitare:

```
$ man mailx
```

Oppure, consultare la guida *man Pages(1): User Commands.*

Uso delle stampanti 8

Il servizio di stampa LP (dal sottosistema *line printer*) è la parte di SunOS che contiene le utility per la stampa. LP offre un'ampia serie di funzioni, molte delle quali non saranno trattate in questo manuale. Questo capitolo descrive le procedure necessarie per eseguire le seguenti operazioni:

- Invio di una richiesta di stampa (invio di un file alla stampante)
- Determinazione dello stato di una stampante
- Annullamento di una richiesta di stampa

Per una descrizione più completa del servizio di stampa LP, vedere il volume *User Accounts, Printers, and Mail Administration*.

Invio di una richiesta di stampa

Per stampare un file dal prompt di sistema si utilizza il comando `lp`, che invia alla stampante la richiesta di stampare il file specificato. Quando si invia una richiesta di stampa, il servizio LP la inserisce nella coda della stampante, visualizza l'ID della richiesta e quindi torna a visualizzare il prompt di sistema.

Invio di una richiesta di stampa alla stampante di default

Se il servizio di stampa LP è configurato con una stampante di default, le richieste di stampa possono essere formulate come segue senza specificare il nome della stampante:

```
$ lp nomefile
```

dove *nomefile* è il nome del file che si desidera stampare.

Il file specificato viene collocato nella coda di stampa della stampante di default e il sistema visualizza l'*id della richiesta*.

Ad esempio, per stampare il file `/etc/passwd`, digitare:

```
$ lp /etc/passwd
l'id della richiesta è pigna-8 (1 file)
$
```

Per informazioni su come specificare una stampante di default, vedere il volume *User Accounts, Printers, and Mail Administration*.

Invio di una richiesta di stampa a una stampante specifica

Indipendentemente dalla designazione o meno di una stampante di default, le richieste di stampa possono essere inviate a qualsiasi stampante configurata per il proprio sistema. Per inviare una richiesta di stampa a una stampante specifica, digitare quanto segue:

```
$ lp -d nomestampante nomefile
```

dove *nomestampante* è il nome della stampante da utilizzare e *nomefile* è il nome del file da stampare.

Il file specificato viene collocato nella coda di stampa della stampante di destinazione e il sistema visualizza l'id della richiesta.

Ad esempio, per stampare il file `/etc/passwd` sulla stampante fungo, digitare:

```
$ lp -d fungo /etc/passwd
l'id della richiesta è fungo-9 (1 file)
$
```

Se la richiesta di stampa viene inviata a una stampante che non è configurata per il proprio sistema, viene visualizzato il messaggio informativo indicato nell'esempio:

```
$ lp -d gnomo /etc/passwd
UX:lp: ERRORE: La destinazione "gnomo" è sconosciuta
               al servizio di stampa LP.
$
```

Per informazioni sulla configurazione delle stampanti, vedere il manuale *User Accounts, Printers, and Mail Administration*. Per informazioni su come visualizzare le stampanti disponibili per il proprio sistema, vedere "Determinazione dello stato della stampante" più avanti in questo capitolo.

Richiesta di notifica al termine della stampa

Quando si invia alla stampante un file molto lungo, può essere utile ricevere da LP una notifica che segnali la fine della stampa. Il servizio di stampa LP può essere impostato in modo da inviare la notifica in una delle forme seguenti:

- Con un messaggio di posta elettronica
- Con un messaggio visualizzato nella finestra della Console

Per richiedere una notifica tramite posta elettronica, usare l'opzione `-m` all'invio della richiesta di stampa:

```
$ lp -m nomefile
```

Per richiedere la visualizzazione del messaggio nella Console, usare l'opzione `-w` all'invio della richiesta di stampa:

```
$ lp -w nomefile
```

dove *nomefile* è il nome del file da stampare.

Stampa di un file in più copie

Un file può essere stampato in una o più copie. Se si desidera stampare più di una copia, specificare il numero di copie desiderato con l'opzione `-n` del comando `lp`. La richiesta di stampa verrà considerata un unico *job*, e verrà stampata solo una pagina di intestazione.

Per richiedere la stampa di più copie di un file, digitare il comando:

```
$ lp -nnumero nomefile
```

dove *numero* è il numero di copie desiderato e *nomefile* è il nome del file da stampare.

Per stampare ad esempio quattro copie del file `/etc/passwd`, digitare:

```
$ lp -n4 /etc/passwd
l'id della richiesta è pigna-9 (1 file)
$
```

Tabella riassuntiva delle opzioni di `lp`

Le richieste di stampa possono essere personalizzate usando le opzioni del comando `lp`: è possibile specificare formati, set di caratteri, filtri, titoli, intestazioni e altri aspetti della stampa. La Tabella 8-1 riassume le opzioni di uso più frequente del comando `lp`. Queste opzioni possono essere utilizzate singolarmente o in una qualsiasi combinazione nella riga di comando. Se si utilizzano più opzioni, separarle con uno spazio e ripetere per ognuna il trattino (`-`).

Ad esempio, per specificare una stampante di destinazione, richiedere una notifica con un messaggio di posta elettronica e stampare sei copie di un file, il comando da inserire sarà il seguente:

```
$ lp -d nomestampante -m -n6 nomefile
```

dove *nomestampante* è il nome della stampante desiderata e *nomefile* è il nome del file da stampare.

Tabella 8-1 Sommario delle opzioni principali di `lp`

Opzione	Descrizione
-d	Specifica la stampante di destinazione.
-m	Invia al richiedente un messaggio di posta elettronica per segnalare il termine della stampa.
-n	Specifica il numero di copie da stampare.
-t	Mostra lo stato del servizio di stampa.
-o nobanner	Sopprime la stampa della pagina di intestazione per una specifica richiesta.
-h	Inserisce un'intestazione su ogni pagina della richiesta di stampa.
-c	Copia il file prima della stampa.
-w	Visualizza un messaggio nella console per segnalare il termine della stampa.

Per un elenco completo delle opzioni disponibili, vedere *man Pages(1): User Commands.*

Determinazione dello stato della stampante

Il comando `lpstat` permette di visualizzare lo stato corrente del servizio di stampa LP. Con questo comando è possibile controllare lo stato dei job nella coda di stampa, identificare le stampanti disponibili o individuare gli id di richiesta dei job per annullarli.

Controllo dello stato delle richieste di stampa

Per visualizzare lo stato delle proprie richieste di stampa, digitare il comando:

```
$ lpstat
```

Viene visualizzato un elenco dei file in coda per la stampa.

Nell'esempio seguente, riferito al sistema `pino`, l'output del comando indica che è presente un file nella coda di stampa della stampante `pigna`:

```
$ lpstat
pigna-10             paola           1261   mar 12 17:34 su pino
$
```

Il comando `lpstat` visualizza una riga per ogni job di stampa; vengono indicati l'id della richiesta, il nome dell'utente che ha inviato la richiesta di stampa, la dimensione dell'output in byte e la data e l'ora della richiesta.

Controllo delle stampanti disponibili

Per determinare quali stampanti sono configurate per il proprio sistema, digitare quanto segue:

```
$ lpstat -s
```

Il sistema visualizza lo stato dello scheduler, la destinazione di default e un elenco dei sistemi e delle stampanti disponibili.

Nell'esempio seguente, riferito al sistema `olmo`, lo scheduler è attivo, la stampante di default è `pigna` e sono disponibili due stampanti in rete, `pigna` e `fungo`:

```
$ lpstat -s
lo scheduler è attivo
destinazione di default del sistema: pigna
sistema per pigna: pino
sistema per fungo: quercia
$
```

Visualizzazione di tutte le informazioni di stato

L'opzione `-t` di `lpstat` visualizza un breve elenco delle informazioni di stato del servizio di stampa LP.

Per visualizzare un breve elenco di tutte le informazioni di stato, digitare:

```
$ lpstat -t
```

Vengono visualizzate tutte le informazioni disponibili sullo stato di LP.

Nell'esempio seguente la coda di stampa è vuota. Quando sono presenti richieste di stampa, il sistema visualizza anche lo stato di tali richieste:

```
$ lpstat -t
lo scheduler è attivo
destinazione di default del sistema: pigna
sistema per fungo: quercia
pigna accetta richieste da giovedì, 3 gennaio 1993, 18:20:10 MET
fungo accetta richieste da lunedì, 4 marzo 1993, 15:53:47 MET
stampante pigna: inattiva, abilitata da giovedì 3 gennaio 1993,
18:20:22 MET. disponibile.

stampante fungo: inattiva, abilitata da lunedì, 4 marzo 1993,
15:53:44 MET. disponibile.
$
```

Visualizzazione dello stato delle stampanti

Per ottenere informazioni sullo stato di una stampante specifica, è possibile utilizzare l'opzione `-p` di `lpstat`. Questa opzione permette di sapere se la stampante è attiva o disattiva, la data e l'ora in cui è stata abilitata o disabilitata e indica se la stampante è disponibile per l'accettazione di richieste di stampa.

Per visualizzare lo stato di tutte le stampanti di un sistema, digitare:

```
$ lpstat -p
```

Nell'esempio seguente, le due stampanti configurate per il sistema sono in stato non attivo, abilitato e disponibile. Se una delle stampanti contenesse uno o più job nella coda di stampa, l'output del comando mostrerebbe anche i job in coda.

```
$ lpstat -p
stampante pigna: inattiva, abilitata da giovedì 3 gennaio 1993,
18:20:22 MET. disponibile.
stampante fungo: inattiva, abilitata da lunedì, 4 marzo 1993,
15:53:44 MET. disponibile
$
```

Per richiedere lo stato di una stampante specifica, digitare il comando:

```
$ lpstat -p nomestampante
```

dove *nomestampante* è la stampante su cui si desidera avere informazioni.

Visualizzazione delle caratteristiche della stampante

Per visualizzare tutte le caratteristiche di una stampante, utilizzare le opzioni `-p` e `-l` di `lpstat`. Questo comando è particolarmente utile per identificare il tipo di stampante e il tipo di contenuto.

Per visualizzare le caratteristiche di tutte le stampanti del sistema, digitare il comando seguente:

```
$ lpstat -p -l
```

Vengono visualizzate in forma di tabella tutte le informazioni di configurazione utilizzate dal servizio di stampa LP per ogni stampante.

Nell'esempio seguente, riferito alla stampante `pigna`, tutti i campi sono vuoti ad eccezione del tipo di contenuto e del tipo di stampante.

```
$ lpstat -p pigna -l
stampante pigna: inattiva, abilitata da giovedì 3 gennaio 1993,
18:20:22 MET. disponibile.
        Tipi di contenuto: PS
        Tipi di stampante: PS
        Descrizione:
        Utenti consentiti:
                (all)
        Formati consentiti:
                (none)
        Intestazione non richiesta
        Set di caratteri:
                (nessuno)
        Interlinea di default:
        Dimensione pagina di default:
$
```

Tabella riassuntiva delle opzioni di `lpstat`

Le varie opzioni del comando `lpstat` permettono di richiedere diversi tipi di informazioni relative allo stato di stampa. La Tabella 8-2 riassume le opzioni di uso più frequente del comando `lpstat`. Queste opzioni possono essere usate singolarmente o in qualsiasi combinazione nella riga di comando. Se si utilizzano più opzioni, separarle con uno spazio e ripetere per ognuna il trattino (`-`).

Ad esempio, per ottenere una lista completa delle informazioni di stato relative a una stampante specifica, digitare il comando:

```
$ lpstat -p nomestampante -l
```

dove *nomestampante* è la stampante di cui si desidera conoscere lo stato.

Tabella 8-2 Sommario delle opzioni principali di `lpstat`

Opzione	**Descrizione**
-a	Indica se le destinazioni di stampa accettano o meno le richieste di stampa.
-c	Visualizza le classi e i rispettivi membri.
-d	Visualizza la destinazione di default.
-f	Visualizza i formati.
-o	Visualizza lo stato dell'output.
`-p` [*lista*][`-D`][`-l`}	Visualizza lo stato delle stampanti.
-r	Stato dello scheduler delle richieste.
-R	Visualizza la posizione del job nella coda di stampa.
-s	Visualizza un sommario delle informazioni di stato.
-S	Visualizza i set di caratteri.
`-u` [*nomeutente*]	Visualizza le richieste inviate dall'utente.
-v	Visualizza i dispositivi.

Per l'elenco completo delle opzioni, vedere *man Pages(1): User Commands*.

Annullamento di una richiesta di stampa

Il comando `cancel` permette di annullare una richiesta di stampa inserita nella coda o già in corso di stampa. È necessario conoscere l'id della richiesta, composto sempre dal nome della stampante, un trattino e il numero della richiesta di stampa. L'id della richiesta viene visualizzato all'invio del comando di stampa. Se non si ricorda l'id della richiesta da annullare, digitare `lpstat` e premere Return. Le richieste di stampa possono essere annullate solo dall'utente che le ha inviate o da un utente connesso come `root` o `lp`.

Annullamento di una richiesta di stampa con indicazione dell'ID

Per annullare una richiesta di stampa, digitare il comando:

```
$ cancel idrichiesta
```

dove *idrichiesta* è l'id della richiesta di stampa desiderata.

Il sistema visualizza un messaggio indicante che la richiesta è stata annullata e inizia la stampa del job successivo nella coda.

Nell'esempio seguente vengono annullate due richieste di stampa:

```
$ cancel pigna-3 pigna-4
richiesta "pigna-3" annullata
richiesta "pigna-4" annullata
$
```

Annullamento di una richiesta di stampa con indicazione della stampante

È anche possibile annullare solo il job in corso di stampa, digitando il nome della stampante anziché l'id della richiesta:

```
$ cancel nomestampante
```

dove *nomestampante* è la stampante a cui è stata inviata la richiesta.

Il sistema visualizza un messaggio indicante che la richiesta è stata annullata e inizia la stampa del job successivo nella coda.

Nell'esempio seguente viene annullata la richiesta in corso di stampa:

```
$ cancel pigna
richiesta "pigna-3" annullata
$
```

L'amministratore del sistema può operare come utente `root` o `lp` e annullare la richiesta in corso di stampa usando il nome della stampante come argomento per il comando `cancel`.

Operazioni in rete 9

Si definisce *rete* un insieme di computer configurati in modo da comunicare tra loro. All'interno di una rete è possibile utilizzare le risorse di altri sistemi operando normalmente sul proprio. È cioè possibile, operando sulla propria workstation, effettuare il login o eseguire comandi remoti che abbiano effetto su altri sistemi.

Questo capitolo tratta i seguenti argomenti:

- Concetti generali sulle reti
- Login in sistemi remoti
- Copia di file da sistemi remoti
- Esecuzione di comandi su sistemi remoti
- Richiesta di informazioni di stato su sistemi remoti

Se il proprio sistema non è collegato a una rete, le informazioni contenute in questo capitolo potranno ugualmente essere utili per considerare e valutare i vantaggi del lavoro in rete.

Concetti generali sulle reti

Un collegamento in rete permette la trasmissione dei dati da un sistema ad un altro. Le reti vengono di norma classificate come LAN (*local area networks* o *reti locali*), operanti su distanze nell'ordine di centinaia di metri; WAN (*wide area networks*), operanti su distanze fino a migliaia di chilometri; o CAN (*campus area networks*), operanti su distanze intermedie.

A un livello superiore si collocano le cosiddette *internetwork*, reti formate da gruppi di reti collegate. Ad esempio, una workstation può essere collegata a una rete installata all'interno dell'edificio e allo stesso tempo a una internetwork che collega la rete locale ad altre LAN simili in altre città. Poiché la differenza tra rete e internetwork non ha effetti rilevanti per l'utente, in questo manuale il termine "rete" verrà utilizzato per indicare indifferentemente entrambi i tipi di collegamento.

I sistemi collegati in rete comunicano tramite un *protocollo di rete*, o linguaggio comune della rete, per garantire che le informazioni vengano trasmesse alle destinazioni appropriate. Per il collegamento tra le reti si utilizzano invece i *protocolli di internetwork*, detti anche *relay*.

Login remoto (`rlogin`)

Il comando `rlogin` permette di effettuare il login su altri sistemi UNIX della propria rete.

Per effettuare il login su un sistema remoto, digitare:

```
$ rlogin nomesistema
```

dove *nomesistema* è il nome del sistema remoto.

Se compare la richiesta di una password, digitare la password per il sistema remoto e premere Return. Se il nome del proprio sistema è contenuto nel file `/etc/hosts.equiv` del sistema remoto, quest'ultimo non richiederà la password di accesso.

```
$ rlogin solitario
Password: (digitare la password)
Last login: Mon Jan 6 09:37:55 from zoo
Sun Microsystems, Inc.   SunOS 5.1      October 1992
(I comandi seguenti vengono eseguiti su solitario.)
$ pwd
/home/aquila
$ logout
Connection closed.
$
```

`rlogin` *senza una directory home*

Nell'esempio precedente, l'utente `aquila` ha effettuato il login in `solitario` nella directory `/home/aquila`, come indicato dal comando `pwd`. Se si effettua il login in un sistema in cui non si dispone di una propria directory home, `rlogin` visualizza un messaggio indicante l'assenza di tale directory home nel sistema remoto; in questo caso, l'utente accede direttamente alla directory radice (`/`) del sistema remoto:

```
$ rlogin fattoria
Password:
No directory! Logging in with home=/
Last login: Fri Jan 3 10:21:59 from zoo
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(I comandi seguenti vengono eseguiti su fattoria.)
$ pwd
/
$ logout
Connection closed.
$
```

`rlogin` *con un altro nome utente*

In alcuni casi può essere necessario effettuare il login in un sistema remoto nelle vesti di un altro utente. Ad esempio, può accadere di lavorare sul sistema di un altro utente (con il nome di login di questo utente) e di dover accedere al proprio sistema con il proprio nome utente. Questo è possibile con l'opzione `-l` del comando `rlogin`. La sintassi del comando è la seguente:

`rlogin` *nomesistema* `-l` *nomeutente*

Nell'esempio seguente l'utente aquila, operando dal sistema zoo, effettua il login nel sistema solitario come utente lupo:

```
$ rlogin solitario -l lupo
Password:
Last login: Wed Jan 8 07:12:25 from zoo
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(I comandi seguenti vengono eseguiti su solitario.)
$ pwd
/home/lupo
$ logout
Connection closed.
$
```

Effettuando il login in un sistema remoto con il nome di un altro utente, si accede alla directory home di quell'utente.

rlogin *in un sistema sconosciuto*

Se si cerca di effettuare il login in un sistema remoto il cui nome non è definito nel proprio sistema, rlogin cerca senza successo il nome del sistema nel database degli host e visualizza il messaggio seguente:

```
$ rlogin straniero
straniero: unknown host
$
```

Chiusura di una connessione stabilita con rlogin

Per terminare una connessione stabilita con rlogin, si utilizza di norma il comando logout alla fine della sessione di lavoro. Se per qualche ragione non fosse possibile terminare la sessione in questo modo, la connessione può essere interrotta digitando una tilde seguita da un punto (~.) all'inizio di una riga. La connessione con il sistema remoto viene interrotta e l'utente ritorna nel sistema di partenza.

Se si effettua il login in una serie di sistemi accedendo ad ognuno attraverso un altro sistema, e si utilizza quindi `~.` per interrompere la connessione con uno qualsiasi di questi sistemi, si ritorna al sistema di partenza:

```
$ rlogin dakota
Password:
Last login: Fri Jan 10 09:14:43 from zoo
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(I comandi seguenti vengono eseguiti su dakota.)
$ ~.   (È possibile che il carattere ~ non appaia sullo schermo.)
Connection closed.
$
```

Se si desidera tornare a una connessione intermedia stabilita con `rlogin`, digitare due caratteri tilde seguite da un punto (`~~.`), come illustrato in questo esempio:

```
$ rlogin solitario
Password:
Last login: Tue Jan 7 08:12:49 from zoo
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(I comandi seguenti vengono eseguiti su solitario.)
$ rlogin dakota
Password:
Last login: Tue Jan 7 10:17:40 from solitario
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(I comandi seguenti vengono eseguiti su dakota.)
$ ~~.   (È possibile che i caratteri ~~ non appaiano sullo schermo.)
Connection closed.
$
```

Sospensione di una connessione stabilita con `rlogin`

Per sospendere temporaneamente una connessione stabilita con `rlogin`, digitare una tilde (~) e quindi premere Ctrl-Z. La connessione stabilita con `rlogin` assume lo stato di processo sospeso e l'utente ritorna al sistema di partenza.

Per riattivare la connessione, digitare `fg`. Oppure, digitare il simbolo di percentuale (%) seguito dal numero del processo sospeso (se non viene specificato alcun numero di processo, viene considerato per default il numero dell'ultimo processo sospeso).

```
$ rlogin solitario
Password:
Last login: Tue Jan 7 08:12:49 from zoo
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(Il comando seguente viene eseguito su solitario.)
 ~^Z   (È possibile che ^Z non compaia sullo schermo.)
Stopped
(Il comando seguente viene eseguito su zoo.)
$ pwd
/home/aquila
$ %
rlogin solitario

(Il comando seguente viene eseguito su solitario.)
$ logout
Connection closed.
$
```

Analogamente alla chiusura della connessione `rlogin` con `~~.`, digitando due caratteri tilde e Ctrl-Z la connessione `rlogin` corrente viene interrotta e l'utente torna a un `rlogin` intermedio.

Verifica della posizione corrente (`who am i`)

Dopo avere effettuato una serie di login in sistemi remoti e avere utilizzato vari nomi di login, può essere utile verificare l'esatta posizione corrente. Il comando `who am i` visualizza il nome del sistema corrente e il proprio nome di login corrente.

Per maggiori informazioni, digitare `man rlogin` al prompt di sistema o consultare la guida *man Pages(1): User Commands.*

Copia remota dei file (rcp)

Il comando `rcp` permette di copiare uno o più file da un sistema a un altro. Esso utilizza i file `/etc/hosts.equiv` e `/etc/passwd` del sistema remoto per determinare i privilegi di accesso dell'utente. La sintassi di `rcp` è analoga a quella del comando `cp`.

Nota – Per copiare una sottodirectory e il suo contenuto da un sistema a un altro, usare il comando `rcp -r`.

Copia di file da un altro sistema al proprio

Per copiare uno o più file da un altro sistema sul proprio, la sintassi del comando è la seguente:

`rcp` *nomesistema*:*sorgente destinazione*

dove *nomesistema* è il nome del sistema remoto, *sorgente* è il nome del file che si desidera copiare e *destinazione* è il percorso in cui si desidera copiare il file sul proprio sistema.

Nell'esempio seguente, il file `/home/dakota/doc/lettera` viene copiato dal sistema remoto `dakota` alla directory `/tmp` del sistema locale `zoo`:

```
$ rcp dakota:/home/dakota/doc/lettera /tmp
$
```

Con il comando `rcp` è anche possibile utilizzare varie combinazioni di abbreviazioni e sintassi. Ad esempio, per copiare tutti i file che terminano in `.doc` dalla directory home dell'utente `gallo` sul sistema remoto `fattoria` alla directory corrente del sistema locale `zoo`, il comando da digitare sarà:

```
$ rcp fattoria:~gallo/*.doc .
$
```

Copia di file dal proprio sistema ad un altro

Per copiare uno o più file dal proprio sistema locale in un sistema remoto, la sintassi del comando viene invertita come segue:

`rcp` *sorgente nomesistema*:*destinazione*

dove *sorgente* è il file che si desidera copiare, *nomesistema* è il nome del sistema remoto e *destinazione* è il percorso in cui si desidera copiare il file nel sistema remoto.

Nell'esempio seguente, il file `asti` viene copiato dalla directory `~/italia/piemonte` del sistema di partenza nella directory `~gallo/province` del sistema remoto `fattoria` (si ricordi che ~ indica la directory home dell'utente che esegue il comando e `~gallo` è la directory home dell'utente `gallo`):

```
$ rcp ~/italia/piemonte/asti fattoria:~gallo/province
$
```

Per maggiori informazioni sul comando `rcp` e sulle relative opzioni, vedere la guida *man Pages(1): User Commands*.

Esecuzione remota dei comandi (`rsh`)

Il comando `rsh` (da *remote shell*) permette di eseguire un comando singolo su un sistema remoto senza bisogno di effettuare il login su quel sistema. Questa possibilità può risultare molto utile quando si desidera effettuare una sola operazione sul sistema remoto.

Per eseguire un comando sul sistema remoto, digitare:

`rsh` *nomesistema comando*

Nell'esempio seguente viene visualizzato il contenuto della directory `/home/solitario/chitarre` del sistema remoto `solitario`:

```
$ rsh solitario ls /home/solitario/chitarre
collings         gibson           santacruz
fender           martin           taylor
$
```

Come i comandi `rlogin` e `rcp`, `rsh` utilizza i file `/etc/hosts.equiv` e `/etc/passwd` del sistema remoto per determinare i privilegi di accesso dell'utente.

Per maggiori informazioni sul comando `rsh` e sulle relative opzioni, vedere la guida *man Pages(1): User Commands*.

Visualizzazione di informazioni sugli utenti (`rusers`)

Il comando `rusers` (da *remote users*) visualizza gli utenti connessi ai vari sistemi della rete. Digitando il comando `rusers` da solo vengono visualizzati tutti i sistemi della rete e gli utenti connessi in ciascuno:

```
$ rusers
bosco          quercia
dakota         sandra
fattoria       gallo
firenze        linda    paola     bruno
foresta        sequoia
pianeti        marte
solitario      lupo
zoo            leone
$
```

Si noti che il sistema `firenze` ha tre utenti connessi.

Per visualizzare le informazioni relative a uno specifico sistema remoto, digitare il comando `rusers` seguito dal nome del sistema, come segue:

```
$ rusers firenze
firenze        linda    paola     bruno
$
```

L'opzione `-l` del comando `rusers` permette di ottenere informazioni più dettagliate, tra cui i nomi degli utenti, i nomi dei sistemi e dei terminali, l'ora di connessione di ogni utente, il tempo di inattività di ogni utente (se maggiore di un minuto) e il nome del sistema da cui ogni utente ha effettuato il login:

```
$ rusers -l firenze
linda        firenze:ttyd8     Feb 10 08:12     5:29
paola        firenze:console   Feb 10 09:16
bruno        firenze:ttyp0     Feb 10 11:56       36
$
```

L'opzione `-l` può essere utilizzata anche senza specificare il nome di un sistema.

Per maggiori informazioni sul comando `rusers` e sulle relative opzioni, vedere la guida *man Pages(1): User Commands.*

Personalizzazione dell'ambiente di lavoro 10

Il sistema operativo SunOS permette all'utente di controllare e configurare direttamente molti aspetti dell'ambiente di lavoro. Questo è possibile con la modifica delle *variabili d'ambiente* contenute nei *file di inizializzazione* del sistema. Quando si effettua il login, il sistema legge i file di inizializzazione e utilizza le variabili d'ambiente per configurare il sistema. Impostando o modificando opportunamente le variabili d'ambiente è perciò possibile "personalizzare" il sistema per adattarlo alle proprie esigenze di lavoro.

Questo capitolo spiega come personalizzare il sistema modificando i file di inizializzazione e impostando le variabili d'ambiente più comuni. Descrive inoltre le procedure da seguire per creare alias per i comandi di SunOS, per modificare il prompt di sistema, impostare le autorizzazioni di default e personalizzare i font di OpenWindows.

File di inizializzazione

I file di inizializzazione che determinano la configurazione del sistema dipendono dal tipo di shell di default specificata dall'amministratore del sistema al momento dell'installazione. La Bourne shell è la shell di default per i sistemi SunOS, ma è anche possibile utilizzare una C shell o una Korn shell. Ad ognuna di queste sono associati uno o più file di inizializzazione.

Se non si è certi della shell di default impostata per il proprio sistema (detta anche *shell di login*), procedere come segue:

1. **Digitare** `echo $SHELL`:

```
$ echo $SHELL
/bin/sh
```

2. **Osservare l'output del comando. Se è:**

- `/bin/sh` – la shell di login è una Bourne shell
- `/bin/csh` – la shell di login è una C shell
- `/bin/ksh` – la shell di login è una Korn shell

Indipendentemente dalla shell utilizzata, quando l'utente effettua il login il sistema esegue il file di profilo del sistema, `/etc/profile`. Questo file è in genere di proprietà dell'amministratore del sistema ed è accessibile in lettura (ma non in scrittura) da tutti gli utenti.

Dopo il profilo del sistema, il sistema esegue il *profilo utente*. Il profilo utente è rappresentato da uno o più file di inizializzazione che definiscono l'ambiente di lavoro. Ad esempio, se ci si trova nell'ambiente OpenWindows il sistema controlla il file (o i file) di inizializzazione ogni volta che viene avviata una Shell o una Finestra di comando.

A seconda della shell di default utilizzata, i file del profilo utente possono essere i seguenti:

- `.profile` (per la Bourne shell e la Korn shell)
- `.login` e `.cshrc` (per la C shell)

I file del profilo utente sono contenuti nella directory home e consentono di configurare l'ambiente di lavoro in base alle proprie preferenze.

Variabili d'ambiente

Il sistema configura l'ambiente di lavoro in base a una serie di impostazioni definite nei file di inizializzazione. Per modificare temporaneamente l'ambiente di lavoro, ad esempio per la sessione corrente, è possibile eseguire i comandi appropriati direttamente al prompt di sistema. Se invece si desidera apportare modifiche di carattere permanente, occorre memorizzare variabili d'ambiente "permanenti" nei file `.profile`, `.login` o `.cshrc`.

Per visualizzare le variabili d'ambiente correntemente impostate sul sistema, procedere come segue:

1. Digitare il comando `env` e premere Return:

```
$ env
HISTORY=100
HOME=/home/italia/logos
HZ=100
LOGNAME=logos
MAIL=/var/mail/logos
MANSECTS=\1:1m:1c:1f:1s:1b:2:\3:3c:3i:3n:3m:3k:3g:3e:3x11:3xt:3
w:3b:9:4:5:7:8
PATH=/usr/bin
SHELL=/bin/sh
TERM=sun
TZ=EST5EDT
```

Nota – È anche possibile usare il comando `env` per identificare la propria shell di login, specificata nella variabile d'ambiente `SHELL`. Nell'esempio precedente, la shell è impostata come `/bin/sh` (Bourne shell).

Profilo utente

Nei paragrafi seguenti vengono descritte alcune delle variabili d'ambiente più utilizzate. Molte di queste variabili potrebbero essere già definite nel proprio profilo utente. Come accennato in precedenza, il file del profilo utente (`.profile` per la Bourne shell e la Korn shell e `.cshrc` per la C shell) si trova nella directory home.

Nota – I file nascosti (file "punto") possono essere visualizzati con il comando `ls -la`.

L'elenco seguente illustra alcune delle variabili d'ambiente che è possibile impostare nel proprio profilo utente. La sintassi per la definizione delle variabili d'ambiente dipende dal tipo di shell in uso:

- `CDPATH` – Specifica le directory in cui il sistema deve cercare quando viene digitato il nome di una directory senza specificare un percorso completo.

- `HISTORY` – Definisce il numero dei comandi memorizzati dal comando `history` (solo per la C shell).
- `HOME` – Definisce il percorso assoluto della directory home dell'utente. Il sistema utilizza questa informazione per determinare la directory in cui spostarsi quando viene digitato il comando `cd` senza argomenti.
- `LANG` – Specifica la lingua locale. Sono ammessi i valori: Japanese, German, French, Swedish e Italian.
- `LOGNAME` – Definisce il nome di login dell'utente. Il valore di default per questa variabile è il nome di login specificato nel database `passwd` come parte del processo di login. Per informazioni sul database `passwd`, vedere il manuale *User Accounts, Printers, and Mail Administration*.
- `LPDEST` – Definisce la stampante di default.
- `MAIL` – Specifica il percorso della casella postale dell'utente. Questa è di norma collocata nella directory `/var/mail/`*nomeutente*, dove *nomeutente* è il nome di login dell'utente. Per maggiori informazioni su questo file, vedere il Capitolo 7, "La posta elettronica".
- `MANSECTS` – Imposta le sezioni disponibili del manuale di riferimento online (pagine `man`).
- `PATH` – Elenca, nell'ordine, le directory in cui il sistema deve cercare per trovare un programma da eseguire quando viene inserito un comando. Se la directory richiesta non si trova nel percorso di ricerca, occorre inserirla nella variabile oppure digitare il percorso completo quando si inserisce il comando.

  L'impostazione di default per questa variabile viene definita automaticamente in base a quanto specificato nel file `.profile` (Bourne shell o Korn shell) o `.cshrc` (C shell) nel processo di login.
- `PS1` – Definisce il prompt di sistema. Il prompt di default per la Bourne shell e la Korn shell è il simbolo del dollaro (`$`). Il prompt di default per la C shell è il simbolo di percentuale (`%`). Il prompt di default per l'utente root in tutte le shell è il simbolo del cancelletto (#).
- `SHELL` – Definisce il tipo di shell utilizzata da `vi` ed altri programmi.

- TERMINFO – Specifica il percorso dei terminali non supportati che sono stati aggiunti al database terminfo. L'impostazione di questa variabile non è necessaria per i terminali di default definiti nel database. Per informazioni sul database terminfo, vedere il manuale *User Accounts, Printers, and Mail Administration*.
- TERM – Definisce il tipo di terminale correntemente in uso. Quando si avvia un editor, il sistema cerca un file con un nome corrispondente alla definizione di questa variabile. Il file viene cercato prima nel percorso eventualmente specificato nella variabile TERMINFO, e quindi nella directory di default, /usr/share/lib/terminfo, per determinare le caratteristiche del terminale. Se il sistema non trova una definizione in nessuna delle directory esaminate, il terminale viene identificato come asincrono.
- TZ – Definisce il fuso orario utilizzato dall'orologio del sistema.

Impostazione della variabile PATH

La variabile d'ambiente PATH viene usata dal sistema per localizzare i comandi nella gerarchia di directory di SunOS. Con la definizione di PATH si stabilisce un insieme fisso di directory in cui il sistema va a cercare ogni volta che viene inserito un comando.

Se la variabile PATH non è impostata e si desidera ad esempio copiare un file, sarà necessario inserire il percorso completo del comando di copia, /usr/bin/cp. Se invece la variabile PATH è stata impostata in modo da includere la directory /usr/bin, sarà sufficiente digitare cp perché il sistema esegua il comando. In questo caso, infatti, il sistema cercherà il comando cp in tutte le directory incluse nella variabile PATH. L'inserimento nella variabile delle directory che contengono i comandi più comuni di SunOS è perciò determinante per semplificare e velocizzare il lavoro.

Se si utilizza una shell di tipo Bourne o Korn, la variabile PATH deve essere specificata nel file .profile (nella directory home) con la sintassi seguente:

```
PATH=.:/usr/bin:/home/bin
```

dove *home* rappresenta il percorso della directory home dell'utente.

Se si utilizza la C shell, la variabile PATH deve essere specificata nel file `.cshrc` (nella directory home) con la sintassi seguente:

```
set path=(. /usr/bin home/bin)
```

dove *home* è il percorso della directory home dell'utente.

Nota – Nelle C shell è anche possibile usare il simbolo ~ per rappresentare il percorso della directory home.

Se si modifica la variabile PATH e si utilizza una C shell, il comando `source` permette di applicare immediatamente le modifiche apportate senza bisogno di effettuare il logout:

```
esempio% source .cshrc
```

Se si utilizza una shell di tipo Bourne o Korn, lo stesso risultato si può ottenere con il comando seguente:

```
$ . .profile
```

Alias (solo C Shell)

Gli alias sono abbreviazioni che sostituiscono comandi di uso frequente. Ad esempio, l'impostazione di default per il comando `rm` non prevede una richiesta di conferma prima di eseguire la cancellazione. Questa impostazione può essere rischiosa, poiché un errore di battitura può causare la cancellazione del file sbagliato. La C shell consente tuttavia di utilizzare la variabile `alias` per modificare l'impostazione di default aggiungendo la riga seguente al file `.cshrc`:

```
alias rm  'rm -i'
```

Con questa istruzione nel file `.cshrc`, il comando `rm` viene reso equivalente a `rm -i`, che rappresenta la sua forma interattiva. In questo modo, il sistema chiederà sempre la conferma dell'utente prima di eseguire la cancellazione di

un file. (Le virgolette che racchiudono `rm -i` nell'esempio sono necessarie per includere lo spazio vuoto tra `rm` e `-i`. Senza le virgolette, la C shell non potrebbe interpretare correttamente i caratteri dopo lo spazio.)

Per applicare immediatamente le modifiche apportate al file `.cshrc`, usare il comando `source`. Questo comando fa in modo che il sistema rilegga il file `.cshrc` ed esegua i relativi comandi:

```
esempio% source .cshrc
```

Modifica del prompt di sistema

La sintassi da utilizzare per cambiare il prompt di sistema dipende dal tipo di shell utilizzata (Bourne, Korn o C shell).

Bourne shell e Korn shell

Se si utilizza una shell di tipo Bourne o Korn, il prompt di sistema può essere ridefinito con il comando `PS1`. L'esempio seguente mostra tre possibili applicazioni del comando:

```
PS1=": "
PS1="‘hostname‘: "
PS1="‘hostname‘{‘id‘}}: "
```

- Il primo esempio imposta il prompt come due punti (:) seguiti da uno spazio.
- Il secondo esempio crea un prompt formato dal nome del sistema seguito da due punti e uno spazio.
- Il terzo esempio definisce un prompt formato dal nome del sistema seguito dal nome di login dell'utente fra parentesi graffe `{}`, due punti e uno spazio.

Si provi a cambiare il proprio prompt di sistema usando uno dei comandi illustrati nell'esempio. Il prompt resterà nella forma impostata fino alla modifica successiva o fino al logout.

Per modificare il prompt in modo permanente, aggiungere una delle righe indicate (o un prompt di propria creazione) al file`.profile`. In questo modo, il prompt specificato apparirà ad ogni login e in ogni shell.

C Shell

Se si utilizza una C shell, il prompt di sistema può essere personalizzato con il comando `set prompt`. L'esempio seguente mostra tre possibili applicazioni del comando:

```
set prompt="% "
set prompt="‘hostname‘\!: "
set prompt="‘hostname‘{‘id‘}}: "
```

- Il primo comando imposta il prompt come un simbolo di percentuale seguito da uno spazio.
- Il secondo comando crea un prompt formato dal nome del proprio sistema seguito dal numero di storia del comando (hostname1, hostname2, hostname3 e così via).
- Il terzo comando crea un prompt formato dal nome del sistema seguito dal nome di login dell'utente fra parentesi graffe, due punti e uno spazio.

Si provi a cambiare il proprio prompt corrente usando uno dei comandi dell'esempio. Il prompt resterà nella forma definita fino alla modifica successiva o fino al logout dal sistema.

Per modificare il prompt in modo permanente, aggiungere una delle righe indicate (o un prompt di propria creazione) al file `.cshrc`. In questo modo, il prompt specificato apparirà ad ogni login e in ogni shell.

Altre variabili

Oltre a quelle descritte finora, l'utente può impostare molte altre variabili nei file `.profile` o `.cshrc`. Per un elenco completo delle variabili, vedere la guida *man Pages(1): User Commands*. Qui di seguito è fornita una breve descrizione di alcune delle opzioni più utili.

L'istruzione `set noclobber` impedisce di sovrascrivere accidentalmente un file durante una copia con il comando `cp`. Questa variabile è disponibile solo se si utilizza una C shell. Inserire nel file `.cshrc` la riga seguente:

```
set noclobber
```

L'istruzione `set history` definisce il numero dei comandi che verranno visualizzati con il comando `history`. La lista visualizzata da `history` può essere utilizzata anche per ripetere comandi precedenti. Questa variabile è disponibile solo se si utilizza una C shell. Per impostarla, inserire la riga seguente nel file `.cshrc`:

```
set history=100
```

Per ottenere lo stesso risultato utilizzando una shell di tipo Bourne o Korn, inserire la riga seguente nel file `.profile`:

```
HISTORY=100
```

Impostazione delle autorizzazioni di default

Il comando `umask` imposta le autorizzazioni di default per i file e le directory di nuova creazione. Ad esempio, per mantenere un certo grado di protezione e assegnare ai membri del proprio gruppo e a tutti gli altri utenti solo le autorizzazioni di lettura ed esecuzione (-rwxr-xr-x) per i propri file e directory, impostare `umask` nei file `.cshrc` e `.profile` in modo da proteggere tutti i file e directory di propria creazione con le autorizzazioni desiderate.

Analogamente al comando `chmod`, `umask` utilizza un codice numerico per rappresentare le autorizzazioni assolute dei file. Tuttavia, il metodo utilizzato da `umask` per calcolare tale codice si differenzia nettamente da quello usato da `chmod`.

In primo luogo, se `umask` è impostato su 000 significa che tutti i file creati dall'utente avranno le seguenti autorizzazioni (lettura e scrittura ma non esecuzione):

`rw-rw-rw-` (modo 666)

mentre tutte le directory avranno le seguenti autorizzazioni (lettura, scrittura ed esecuzione):

`rwxrwxrwx` (modo 777)

Per determinare il valore da utilizzare con umask, occorre sottrarre il valore delle autorizzazioni desiderate (il valore che occorrerebbe specificare per il comando chmod) dalle autorizzazioni di default correntemente assegnate ai file. Il risultato ottenuto sarà il valore da utilizzare per il comando umask.

Ad esempio, si supponga di voler cambiare le autorizzazioni di default per i propri file da 666 (rw-rw-rw-) a 644 (rw-r--r--). Sottrarre 644 da 666. Il risultato, 022, è il valore numerico da utilizzare con umask come segue:

```
umask 022
```

Come nel codice numerico utilizzato per il comando chmod, i tre numeri usati con umask hanno i seguenti significati:

- La prima cifra controlla le autorizzazioni del proprietario
- La seconda controlla le autorizzazioni del gruppo
- La terza controlla le autorizzazioni di tutti gli altri utenti

La Tabella 10-1 illustra le autorizzazioni dei file per ogni cifra del codice numerico utilizzato da umask.

*Tabella 10-1*Codici delle autorizzazioni utilizzate da umask

codice umask	Autorizzazioni
0	rwx
1	rw-
2	r-x
3	r--
4	-wx
5	-w-
6	--x
7	---(nessuna)

Per maggiori informazioni sul comando umask, vedere la guida *man Pages(1): User Commands*.

Personalizzazione dei font in OpenWindows

Se si desidera, è possibile personalizzare il tipo e la dimensione dei font utilizzati dalle applicazioni in OpenWindows. I paragrafi seguenti descrivono le procedure da seguire per la personalizzazione dei font.

Definizione del tipo di font e della dimensione

Il font di default utilizzato nelle finestre è il Lucida Sans a 12 punti (`medium`); il font di default per le intestazioni delle finestre è il Lucida Sans Bold. Se si preferisce, è possibile specificare un altro tipo di carattere o un'altra dimensione sia per le finestre che per le intestazioni. La modifica può essere effettuata in forma temporanea, per una singola finestra, oppure in modo permanente per tutte le applicazioni, utilizzando la finestra "Caratteristiche area di lavoro". I paragrafi seguenti descrivono entrambe le procedure.

Font a spaziatura fissa e proporzionale

I font si suddividono in due categorie generali — *a spaziatura fissa* e *a spaziatura proporzionale*. Nei font a spaziatura fissa ogni carattere occupa lo stesso spazio nella riga, mentre nei font a spaziatura proporzionale i caratteri occupano uno spazio variabile a seconda della larghezza. In generale, i font a spaziatura proporzionale risultano esteticamente più gradevoli. Tuttavia, per alcune applicazioni (come le Finestre di comando, le Shell e `xterm`, un diffuso emulatore di terminale) è preferibile l'utilizzo dei font a spaziatura fissa.

Scelta tra font a spaziatura fissa e proporzionale

Il font di default utilizzato nelle Finestre di comando e nelle Shell è un font a spaziatura proporzionale. Nonostante l'aspetto gradevole, nelle finestre utilizzate come terminali i font a spaziatura proporzionale possono causare problemi di allineamento dei caratteri (ad esempio nelle spaziature e nelle tabulazioni). In questo caso è perciò preferibile scegliere un font a spaziatura fissa. Negli esempi seguenti, le finestre utilizzate come terminali impiegano font a spaziatura fissa, mentre le altre finestre e intestazioni adottano font a spaziatura proporzionale.

Definizione del font per una singola finestra

Questo paragrafo descrive la procedura da seguire per modificare il tipo e la dimensione del font per una singola applicazione. Si noti che le modifiche non hanno effetto sulle finestre già aperte; per visualizzare il nuovo font occorre avviare una nuova sessione dell'applicazione.

La sintassi base del comando, illustrata nell'esempio seguente, contiene il nome dell'applicazione, l'opzione `-fn` (font name) e il tipo e la dimensione del font. La "e commerciale" (&) fa ricomparire il prompt di sistema dopo l'invio del comando, rendendo la finestra immediatamente disponibile.

```
$ applicazione -fn tipofont-dimensione &
```

Gli esempi che seguono illustrano l'uso di questo comando per aprire un'applicazione specificando il font da utilizzare e la dimensione dei caratteri.

- L'esempio seguente avvia una Finestra di comando con il font a spaziatura proporzionale `Lucida Sans Typewriter Bold`.

  La dimensione del carattere non è specificata; viene perciò utilizzata la dimensione di default (12 punti).

```
$ cmdtool -fn lucidasans-typewriter-bold &
```

- L'esempio seguente avvia una Shell con il font `Lucida Sans Typewriter Bold` e aumenta la dimensione dei caratteri da 12 a 14 punti.

  Modificando la dimensione del font cambia anche la dimensione della finestra.

```
$ shelltool -fn lucidasans-typewriter-bold-14 &
```

- L'esempio seguente avvia una finestra di terminale `xterm` con il font `terminal-bold` a 16 punti:

```
$ xterm -fn terminal-bold-16 &
```

- L'esempio seguente avvia una finestra di Editor con il font Helvetica Bold a 14 punti:

```
$ textedit -fn helvetica-bold-14 &
```

L'opzione -fn può essere utilizzata con qualsiasi applicazione per specificare il font e la dimensione dei caratteri. Proseguendo nel capitolo, il paragrafo "Elenco dei font disponibili" spiegherà come visualizzare tutti i font disponibili per le applicazioni di OpenWindows.

Assegnazione permanente dei font

Se si utilizza spesso un font personalizzato, può essere utile aggiungerlo come opzione al menu Area di lavoro. Questa operazione può essere eseguita utilizzando la categoria "Menu Programmi" della finestra "Caratteristiche area di lavoro". In questo modo non sarà più necessario digitare ogni volta le opzioni di personalizzazione dalla riga di comando. Se ad esempio si preferisce utilizzare Editor con un carattere più grande, è possibile aggiungere al menu Programmi la seguente riga di comando:

```
textedit -fn lucidasans-typewriter-14
```

È anche possibile aggiungere al menu Programmi varie opzioni relative a una stessa applicazione, in modo da poterla avviare con diversi tipi di font o con caratteri di diverse dimensioni. Per avere ad esempio la possibilità di avviare Editor con caratteri a 12, 14 o 18 punti, si potranno aggiungere al menu Programmi i comandi seguenti:

```
textedit -fn lucidasans-typewriter-12
textedit -fn lucidasans-typewriter-14
textedit -fn lucidasans-typewriter-18
```

Personalizzando in questo modo il menu Programmi dalla finestra "Caratteristiche area di lavoro", sarà possibile avviare Editor con il carattere desiderato selezionando semplicemente l'opzione appropriata dal menu.

Nota – Le righe di comando aggiunte al menu Programmi non devono essere seguite da una “e commerciale” (&).

Visualizzazione dei font disponibili

Se si desidera provare ad utilizzare font diversi da quelli illustrati negli esempi precedenti nelle applicazioni di OpenWindows può essere utile visualizzare l’elenco dei font disponibili.

Elenco dei font disponibili

Per ottenere l’elenco completo dei font disponibili, digitare il comando `xlsfonts` al prompt di sistema da una finestra di emulazione terminale. È preferibile utilizzare una Finestra di comando, in quanto la lista potrebbe scorrere oltre i margini della finestra e richiedere l’utilizzo di una barra di scorrimento.

Nota – L’elenco generato da `xlsfonts` è molto lungo, in quanto contiene i nomi di oltre 400 font. Se l’elenco visualizzato contiene un numero di font relativamente ridotto, consultare l’amministratore di sistema. È possibile che sia stato installato soltanto un sottoinsieme dei font disponibili.

Ogni font possiede un nome completo e uno abbreviato. Il nome completo del font `lucidasans-typewriter`, ad esempio, è:

```
-b&h-lucida sans typewriter-medium-r-normal-sans-12-120-72-72-m-
0-iso8859-1
```

I font visualizzati nell’elenco `xlsfonts` sono presentati con il nome completo seguito dalla rispettiva forma abbreviata. Per le operazioni descritte in questo capitolo sarà sufficiente utilizzare la versione abbreviata.

Dopo avere selezionato un font, seguire le istruzioni del precedente paragrafo “Definizione del tipo di font e della dimensione” per personalizzare i font nelle finestre delle applicazioni.

Migrazione a OpenWindows 3.3 o a versioni successive

In alcuni casi è possibile che l'interfaccia utente installata nel sistema non sia aggiornata per Solaris, che utilizza OpenWindows come interfaccia utente di default. Ad esempio, è possibile che il sistema utilizzi l'ambiente SunView o una versione di OpenWindows anteriore alla 3.3. In questi casi può essere opportuno passare a OpenWindows versione 3.3 o successive. Questa appendice descrive le procedure da seguire.

Nota – Il software SunView non è più supportato in OpenWindows versione 3.3 o successive. A differenza delle precedenti versioni di OpenWindows, installando la versione 3.3 o quelle successive l'interfaccia utente SunView non sarà più disponibile.

SPARC *Migrazione dall'ambiente SunView*

I paragrafi seguenti contengono informazioni e suggerimenti diretti a facilitare la migrazione a OpenWindows dall'ambiente SunView.

SPARC *I file* `.defaults` *e* `.Xdefaults`

Per personalizzare l'ambiente OpenWindows con le stesse opzioni utilizzate in SunView, è possibile convertire il file `.defaults` (usato da SunView) in un file `.Xdefaults` (usato da OpenWindows). Se la propria directory home contiene il file `.defaults`, eseguire il programma `convert_to_Xdefaults(1)` nella directory home come segue:

```
$ cd
$ $OPENWINHOME/bin/convert_to_Xdefaults .defaults
```

Questo comando crea nella directory home il file `.Xdefaults`, utilizzato per personalizzare l'ambiente OpenWindows all'avvio.

Migrazione da una versione di OpenWindows anteriore alla 3.3

Per migrare da una versione di OpenWindows anteriore alla 3.3 alla versione 3.3 o a una successiva, leggere attentamente i paragrafi seguenti. In particolare, molte delle informazioni contenute interessano gli utenti di OpenWindows versione 2.

La variabile d'ambiente `OPENWINHOME`

Se attualmente si utilizza una versione di OpenWindows anteriore alla 3.3, è possibile che il sistema sia stato configurato in modo da utilizzare la variabile d'ambiente `OPENWINHOME`. Con la nuova versione non occorre più impostare questa variabile d'ambiente, né manualmente né in un file di avvio.

In questa versione, infatti, il comando `openwin` imposta automaticamente la variabile d'ambiente `OPENWINHOME` su `/usr/openwin`.

Se la variabile d'ambiente `OPENWINHOME` è impostata nel file `.profile` o `.cshrc` nella directory home, commentare la riga o cancellarla completamente prima di avviare OpenWindows 3.3 o una versione successiva.

Per cancellare o commentare la variabile d'ambiente `OPENWINHOME` nel file `.profile` o `.cshrc`, procedere come segue:

1. **Aprire il file `.profile` o `.cshrc` con un editor di testo, ad esempio `vi`.**

2. **Digitare un cancelletto (#) davanti alla variabile, come illustrato nell'esempio seguente, o cancellare l'intera riga.**
   Per modificare il file `.profile`, seguire l'esempio `a`; per modificare il file `.cshrc`, seguire l'esempio `b`

   a. **Nel file** `.profile`**:**

```
#OPENWINHOME=/usr/openwin
```

   a. **Nel file** `.cshrc`**:**

```
#setenv OPENWINHOME /usr/openwin
```

3. **Salvare il file e uscire dall'editor.**

Il file `.xinitrc`

Questo paragrafo contiene alcune considerazioni importanti relative ai file `.xinitrc` e `$OPENWINHOME/lib/Xinitrc`:

1. In OpenWindows versione 2, lo script `openwin` creava automaticamente una copia di `$OPENWINHOME/lib/Xinitrc` in un file chiamato `.xinitrc`, collocato nella directory home. Questa copia automatica non viene più eseguita nell'ambiente OpenWindows versione 3.3. Questa differenza è importante in quanto:

   a. Lo script di avvio `openwin` utilizza il file di avvio di default, `$OPENWINHOME/lib/Xinitrc`, a meno che non trovi un file di nome `.xinitrc` nella directory home; in questo caso il file `.xinitrc` prevale su quello di default.

   b. È necessario utilizzare il file di default `$OPENWINHOME/lib/Xinitrc` fornito insieme a OpenWindows 3.3 o alle versioni successive. (Se si desidera conservare speciali modifiche apportate al file `.xinitrc` della versione 2, seguire le istruzioni fornite in questo capitolo.)

2. Se si utilizzano più schermi, non è necessario eseguire più processi `olwm`.

Uso del file di avvio corretto

Se attualmente si utilizza una versione di OpenWindows anteriore alla 3.3, è importante determinare lo stato del file `.xinitrc`, collocato nella directory home. Il file `.xinitrc` è un file di avvio di OpenWindows che può contenere opzioni definite dall'utente.

Per determinare lo stato del file `.xinitrc`, digitare i comandi seguenti:

```
$ cd
$ ls -a .xinitrc
```

A seconda dell'output di questo comando, procedere in uno dei modi seguenti:

- Se non è presente un file `.xinitrc` (se cioè l'elenco ottenuto con il comando `ls -a` non contiene il nome di questo file) non fare nulla. Se la propria directory home non contiene il file `.xinitrc`, OpenWindows utilizza il file di avvio di default del sistema.
- Se il file `.xinitrc` è presente (se cioè l'elenco ottenuto con il comando `ls -a` contiene il nome del file) ma non è mai stato modificato, o comunque non si desidera conservare le modifiche apportate, procedere come descritto al punto 1 in "Operazioni sui file di avvio".
- Se il file `.xinitrc` è presente (se cioè l'elenco ottenuto con il comando `ls -a` contiene il nome del file), è stato modificato e si desidera conservare le modifiche apportate, procedere come descritto al punto 2 in "Operazioni sui file di avvio".

Operazioni sui file di avvio

1. **Per cancellare il file** `.xinitrc` **dalla propria directory home, digitare il comando seguente:**

```
$ rm .xinitrc
```

2. **Per conservare le modifiche apportate al file**`.xinitrc`**, procedere come segue:**

a. **Rinominare** `.xinitrc` **in** `.xinitrc.save`:

```
$ mv .xinitrc .xinitrc.save
```

b. **Copiare** `$OPENWINHOME/lib/Xinitrc` **in** `.xinitrc` **nella propria directory home:**

```
$ cp $OPENWINHOME/lib/Xinitrc $HOME/.xinitrc
```

c. **Aggiungere le righe di** `.xinitrc.save` **che si desidera conservare in**`.xinitrc`**.**

Attenzione – Durante la modifica del file `.xinitrc`, non aggiungere una versione secondaria di `olwm`, non aggiungere `svenv` e non cancellare la riga che contiene `$OPENWINHOME/lib/openwin-sys`.

Caratteristiche dell'area di lavoro

Nelle precedenti versioni di OpenWindows (anteriori alla 3.3), le modifiche apportate dalla finestra "Caratteristiche area di lavoro" venivano salvate nel file `.Xdefaults`, collocato nella directory home. Nella versione 3.3 e successive, le modifiche apportate dalla finestra "Caratteristiche area di lavoro" vengono memorizzate nel file `.OWdefaults`, anch'esso collocato nella directory home dell'utente. Il file `.Xdefaults` può ancora essere presente, ma viene data la precedenza alle opzioni personalizzate impostate in `.OWdefaults`.

Il file `.Xdefaults` dovrebbe essere utilizzato *solo* per personalizzare gli aspetti che non possono essere modificati dalla finestra "Caratteristiche area di lavoro". Ad esempio, si potrà aprire il file `.Xdefaults` con un editor, ad esempio `vi`, per personalizzare le applicazioni non-OpenWindows o per aggiungere le macro del preprocessore C. Le opzioni selezionate dalla finestra "Caratteristiche area di lavoro" non influiranno su queste impostazioni personalizzate.

Se la directory home contiene un file `.Xdefaults` e non si desidera modificare le impostazioni correnti, non è necessario cancellarlo. Prevarranno in ogni caso le impostazioni definite nel file `.OWdefaults`.

Personalizzazione del menu Area di lavoro

In OpenWindows 3.3 e nelle versioni successive, l'utente ha la possibilità di personalizzare il sottomenu Programmi del menu "Area di lavoro" usando la finestra "Caratteristiche area di lavoro". Nelle versioni precedenti di OpenWindows, questa personalizzazione richiedeva la modifica del file `.openwin-menu` nella directory home.

Nota – Se la directory home non contiene un file `.openwin-menu` non è necessario eseguire la procedura qui descritta. È possibile personalizzare direttamente il menu "Area di lavoro" usando la finestra "Caratteristiche area di lavoro".

Se la directory home contiene il file `.openwin-menu`, per personalizzare il menu "Area di lavoro" dalla finestra "Caratteristiche area di lavoro" è necessario eseguire la procedura seguente.

Se il file `.openwin-menu` contiene la riga seguente:

```
"Programmi"MENU $OPENWINHOME/lib/openwin-menu-programs
```

cancellarla e sostituirla con la riga seguente:

```
"Programmi"INCLUDE openwin-menu-programs
```

Se il file `.openwin-menu` non contiene questa riga, aggiungere semplicemente la riga sostitutiva indicata.

La nuova riga inserita aggiunge il menu Programmi di default al menu "Area di lavoro". In questo modo il menu Programmi potrà essere personalizzato utilizzando la finestra "Caratteristiche area di lavoro".

Se il menu "Area di lavoro" dovesse contenere voci ridondanti, sarà sufficiente cancellare le righe ridondanti dal file `.openwin-menu`.

Personalizzazione della tastiera

Questa appendice contiene le istruzioni per rimappare la tastiera. Vengono descritte le opzioni necessarie per rimappare alcuni tasti speciali e la procedura per abilitare e disabilitare il tasto Compose.

Per informazioni su come rimappare i bottoni del mouse (ad esempio per utenti mancini), consultare il Manuale utente di Solaris.

Disabilitazione/abilitazione del tasto Compose

x86 – Sui sistemi x86, la funzione del tasto Compose si ottiene premendo Ctrl-Shift-F1.

Se il tasto Compose non viene utilizzato, è possibile disabilitarlo in modo da renderlo inattivo. Per prima cosa, identificare il codice del tasto per Multi_key:

```
$ xmodmap -pk | grep Multi_key
```

Il sistema visualizza una riga simile alla seguente:

nn 0xff20 (Multi_key)

L'informazione importante è il codice a due cifre all'inizio della riga, qui rappresentato con *nn*. Usando questo numero, creare la riga seguente nel file `.xinitrc`:

```
xmodmap -e 'keycode nn = NoSymbol'
```

Per riabilitare il tasto Compose, commentare la riga precedente nel file .xinitrc e riavviare OpenWindows.

SPARC *Rimappatura della tastiera per utenti mancini*

Lo script di rimappatura descritto in questo paragrafo (riferito a tastiere Tipo 4 e Tipo 5) rimappa la maggior parte dei tasti speciali posti a sinistra e a destra del corpo centrale della tastiera.

SPARC – Si noti che i paragrafi seguenti relativi alla rimappatura della tastiera per utenti mancini riguardano solo i sistemi SPARC.

SPARC *Uso dello script di rimappatura*

Per creare e utilizzare lo script di rimappatura, procedere come segue:

1. **Creare un file di nome `lefty.data` usando un editor di testo.**
   Il file può essere creato in qualsiasi directory. Tuttavia, il punto 4 deve essere eseguito nella stessa directory in cui è stato creato questo file.

2. **Inserire lo script come illustrato nella Tabella B-1 "Lo script** lefty.data**".**
   Le righe che iniziano con un punto esclamativo sono commentate, perciò non eseguono alcuna operazione.

3. **Salvare le modifiche e uscire dall'editor.**

4. **Al prompt, digitare:** xmodmap lefty.data
   Questa operazione deve essere eseguita nella directory che contiene il file dello script.

5. **Fare clic con il mouse nell'area di lavoro per avviare lo script.**
   Al termine di questa procedura, i tasti saranno mappati per gli utenti mancini.

Digitare lo script seguente nel file lefty.data, come descritto al punto 1.

Tabella B-1Lo script lefty.data

```
!
! lefty.data
!
! Dati per xmodmap per la configurazione dei tasti funzione di sinistra e di destra per
! utenti mancini su tastiere Sun type-4. Per usare questi dati, digitare il comando
! seguente sostituendo <nomefile> con il nome del file (lefty-data).
!
! xmodmap <nomefile>
!
! Le righe seguenti si riferiscono alle assegnazioni dei codici riportate
! successivamente.
!
! L2 (Ancora)viene scambiato con R1 (Pausa)
! L3 (Proprietà)viene scambiato con R6 (KP_Multiply)
! L4 (Annulla)viene scambiato con R4 (KP_Equal)
! L5 (Davanti)viene scambiato con R9 (KP_9)
! L6 (Copia)viene scambiato con R7 (KP_7)
! L7 (Apri)viene scambiato con R12 (KP_6)
! L8 (Incolla)viene scambiato con R10 (Sinistra)
! L9 (Trova)viene scambiato con R15 (KP_3)
! L10 (Taglia)viene scambiato con R13 (KP_1)
!
! R3 (Break)diventa L1 (Stop)
! R2 (Stampa)diventa R10 (Sinistra)
! R5 (KP_Divide) diventa R12 (Destra)
!
! Linefeed diventa Control-R
!
```

Tabella B-1Lo script lefty.data

```
keycode 10 = R1R1Pause
keycode 28 = L2L2SunXK_Again
keycode 32 = R6R6KP_Multiply
keycode 54 = L3L3SunXK_Props
keycode 33 = R4R4KP_Equal
keycode 52 = L4L4SunXK_Undo
keycode 56 = R9R9KP_9        Prior
keycode 77 = L5L5SunXK_Front
keycode 58 = R7R7KP_7        Home
keycode 75 = L6L6SunXK_Copy
keycode 79 = RightR12KP_6
keycode 100 = L7L7SunXK_Open
keycode 80 = LeftR10KP_4
keycode 98 = L8L8SunXK_Paste
keycode 102 = R15R15KP_3        Next
keycode 121 = L9L9SunXK_Find
keycode 104 = R13R13KP_1        End
keycode 119 = L10L10SunXK_Cut
keycode 30 = L1L1SunXK_Stop
keycode 29 = LeftR10KP_4
keycode 53 = RightR12KP_6

keycode 118 = Control_R
add control = Control_R
```

SPARC *Ripristino della mappatura originale*

Per riportare i tasti alle impostazioni originarie è possibile procedere in due modi. Un primo metodo consiste nell'uscire da OpenWindows e riavviarlo. Il secondo metodo, certamente preferibile se le funzioni dei tasti devono essere commutate periodicamente, consiste nel creare un secondo script da eseguire

ogni volta che si desidera tornare alla mappatura originaria.

Per creare questo secondo script, procedere come segue:

1. **Creare un file di nome** nolefty.data **usando un editor di testo.**
   Il file deve trovarsi nella stessa directory che contiene lo script `lefty.data`.

2. **Inserire lo script come illustrato nella Tabella B-2 "Lo script** nolefty.data**".**
   Le righe precedute da un punto esclamativo sono commentate, perciò non eseguono alcuna operazione.

3. **Salvare le modifiche e uscire dall'editor.**

4. **Al prompt, digitare:**

```
$ xmodmap nolefty.data
```

   Per attivare nolefty.data, digitare il comando precedente nella directory che contiene lo script.

Tabella B-2Lo script nolefty.data

```
!
! nolefty.data
!
! Dati per xmodmap per ripristinare i tasti funzione di sinistra e di destra dopo la
! rimappatura per utenti mancini di una tastiera Sun Tipo 4. Per usare questi dati,
! digitare il comando seguente sostituendo <nomefile> con il nome di questo file.
!
! xmodmap <nomefile>
!
!Riassegna i valori standard ai tasti funzione di sinistra
!
keycode 10 = L2L2SunXK_Again
keycode 32 = L3L3SunXK_Props
keycode 33 = L4L4SunXK_Undo
```

Tabella B-2Lo script nolefty.data

```
keycode 56 = L5L5SunXK_Front
keycode 58 = L6L6SunXK_Copy
keycode 79 = L7L7SunXK_Open
keycode 80 = L8L8SunXK_Paste
keycode 102 = L9L9SunXK_Find
keycode 104 = L10L10Sun XK_Cut
!
! Riassegna i valori standard ai tasti funzione di destra
!
keycode 28 = R1R1Pause
keycode 29 = R2R2Print
keycode 30 = R3R3Scroll_Lock    Break
keycode 52 = R4R4KP_Equal
keycode 53 = R5R5KP_Divide
keycode 54 = R6R6KP_Multiply
keycode 75 = R7R7KP_7           Home
keycode 77 = R9R9KP_9           Prior
keycode 98 = LeftR10KP_4
keycode 100 = RightR12KP_6
keycode 119 = R13R13KP_1        End
keycode 121 = R15R15KP_3        Next
!
! Riassegna la funzione originaria al tasto Linefeed e lo rimuove dalla mappa di
! controllo.
!
remove control = Control_R
5keycode 118 = Linefeed
```

Uso delle applicazioni in rete

Questa appendice descrive una funzione avanzata dell'ambiente OpenWindows che consente di eseguire applicazioni residenti in un'altra workstation della rete.

Nota – Le informazioni contenute in questa appendice non sono strettamente necessarie per l'uso di OpenWindows. Per avere maggiori informazioni sulle caratteristiche delle applicazioni in rete e sulle risorse disponibili, rivolgersi all'amministratore del sistema.

In generale, nell'ambiente OpenWindows le applicazioni che compaiono sullo schermo (come Posta Elettronica e Calendario) sono programmi eseguiti sulla workstation locale. Se tuttavia la workstation fa parte di una rete, è possibile eseguire le applicazioni da un altro sistema e visualizzarle sul sistema locale. Questo permette di ridurre i cicli di calcolo sul sistema locale e di accedere a tutte le applicazioni disponibili nella rete.

Questa appendice descrive le procedure più semplici da seguire per eseguire un'applicazione su un sistema remoto e visualizzarla sul proprio schermo. Poiché l'ambiente operativo può variare da un sistema all'altro, si consiglia di seguire queste istruzioni con una certa flessibilità. Per informazioni sulle modalità d'uso più complesse delle applicazioni in rete, vedere il paragrafo "Sicurezza della rete".

Per eseguire un'applicazione remota nel modo descritto sono necessari i seguenti requisiti:

- È necessario avere diritto d'accesso alla macchina remota.
- La propria directory home deve poter essere attivata tramite NFS sul sistema remoto.
- L'applicazione e le librerie richieste devono essere installate sul sistema remoto, cioè sul sistema *host*.

Per informazioni più precise sulle condizioni richieste, rivolgersi all'amministratore del sistema.

Uso di `rlogin` *per eseguire un'applicazione in rete*

Per eseguire un'applicazione in rete da un sistema remoto è essenziale che le variabili d'ambiente siano impostate correttamente:

- La variabile d'ambiente `HOME` nella propria shell sul sistema remoto deve essere impostata sulla propria directory home.
- La variabile d'ambiente `DISPLAY` nella propria shell sul sistema remoto deve essere impostata sul proprio schermo locale.
- Se le librerie di OpenWindows non sono state installate nelle directory standard per le librerie condivise, vale a dire `/usr/lib` o `/usr/local`, occorre impostare la variabile d'ambiente `LD_LIBRARY_PATH` sulla directory appropriata (`$OPENWINHOME/lib`).

L'esempio seguente spiega come avviare una Finestra di comando da un sistema remoto usando `rlogin`. In questo esempio, la directory home dell'utente è attivata sul sistema remoto in `/home/directory_utente`, mentre il software di OpenWindows si trova in `/usr/openwin` sul sistema remoto. Le variabili *directoryutente* e *sistemautente* devono essere impostate nel modo appropriato per la propria configurazione.Inoltre, cmdtool deve essere sostituito con il nome dell'applicazione che si desidera eseguire.

```
$ rlogin sistemaremoto
    .
    .
(I comandi seguenti vengono eseguiti sul sistema remoto.)]
    .
    .
$ HOME=/home/directoryutente
$ DISPLAY=sistemautente:0
$ LD_LIBRARY_PATH=/usr/openwin/lib
$ /usr/openwin/bin/cmdtool &
```

Dopo aver inserito l'ultima riga, sul proprio schermo compare una Finestra di comando. Questa finestra può essere utilizzata in modo interattivo come qualsiasi altra applicazione aperta nell'area di lavoro, ma viene eseguita in realtà sul sistema remoto.

L'uso di una Finestra di comando in questo modo non presenta particolari vantaggi (l'applicazione è già disponibile sul sistema locale e non richiede una grande quantità di memoria); la procedura descritta può essere comunque utilizzata per qualsiasi applicazione remota.

Sicurezza della rete

Questo paragrafo descrive alcuni principi fondamentali riguardanti la sicurezza della rete, tra cui:

- Meccanismi di controllo degli accessi riferiti all'utente e all'host
- I protocolli di autorizzazione MIT-MAGIC-COOKIE-1 e SUN-DES-1
- Come e quando modificare il controllo degli accessi al server
- Come eseguire le applicazioni in modalità remota o locale con un diverso identificativo utente.

Condizioni per la modifica del sistema di sicurezza

La configurazione di sicurezza di default di OpenWindows 3.3 e delle versioni successive deve essere modificata solo nei seguenti casi:

- Se si esegue un'applicazione collegata con versioni di `Xlib` o `libcps` *anteriori a* OpenWindows versione 2 o a `X11R4`.

- Se si esegue un'applicazione collegata in modo statico con librerie di OpenWindows versione 2 e si desidera utilizzare il protocollo di autorizzazione `SUN-DES-1`.
- Se si esegue un'applicazione su un server remoto.

Meccanismi di controllo degli accessi

I meccanismi di controllo degli accessi permettono di stabilire quali *client*, o applicazioni, possono avere accesso al server X11. Solo i client provvisti delle autorizzazioni corrette possono accedere al server; a tutti gli altri l'accesso viene negato, e il processo viene terminato con un messaggio di errore.

I meccanismi di controllo degli accessi si dividono in due tipi: *riferiti all'utente* e *riferiti all'host*. Nel primo caso l'accesso al server viene concesso a uno specifico profilo utente, nel secondo caso a un determinato *host* o sistema. Di norma sono attivi entrambi i meccanismi di controllo, salvo i casi in cui il comando `openwin` viene eseguito con l'opzione `-noauth`. Per maggiori informazioni, vedere "Gestione degli accessi al server" in questo capitolo.

Accesso riferito all'utente

Con i meccanismi riferiti all'utente, l'accesso al server viene concesso a un determinato utente indipendentemente dall'host utilizzato. Il client dell'utente comunica i dati di autorizzazione al server, e se questi corrispondono a quelli impostati sul server, l'utente ottiene il diritto di accesso.

Accesso riferito all'host

I meccanismi riferiti all'host operano con criteri più generali. In questo caso l'accesso al server viene concesso a un determinato host, e di conseguenza a tutti gli utenti che operano su tale host. Si tratta di una forma di controllo più debole: se l'host ha accesso al server, l'autorizzazione viene estesa automaticamente a tutti gli utenti che lo utilizzano.

I meccanismi di controllo riferiti all'host vengono usati generalmente per garantire la compatibilità all'indietro. Le applicazioni collegate con versioni di `Xlib` o `libcps` anteriori a OpenWindows versione 2 o a X11R4 non riconoscono i nuovi meccanismi di controllo riferiti all'utente. Per ottenere

l'accesso al server con queste applicazioni, perciò, l'utente deve attivare il meccanismo riferito all'host oppure rieffettuare il collegamento con una versione più recente di `Xlib` e `libcps`.

Nota – Se possibile, i client collegati con una vecchia versione di `Xlib` o `libcps` dovrebbero essere ricollegati con versioni più aggiornate delle librerie, in modo da consentire la loro connessione al server con il nuovo meccanismo di controllo riferito all'utente.

Protocolli di autorizzazione

Questa versione di OpenWindows supporta due protocolli di autorizzazione: `MIT-MAGIC-COOKIE-1` e `SUN-DES-1`. Questi due protocolli utilizzano un meccanismo di controllo simile ma un tipo diverso di dati di autorizzazione. Il server può utilizzare un solo protocollo alla volta. Il protocollo di default di OpenWindows è `MIT-MAGIC-COOKIE-1` con meccanismo di protezione riferito all'utente.

`MIT-MAGIC-COOKIE-1`

Il protocollo di autorizzazione `MIT-MAGIC-COOKIE-1` è un prodotto sviluppato dal Massachusetts Institute of Technology. All'avvio del server viene creato un *magic cookie* per il server e per l'utente che ha avviato il sistema. Ad ogni richiesta di connessione, il client dell'utente invia il magic cookie al server come parte del pacchetto di connessione. Il magic cookie inviato viene confrontato con quello del server. Se i due magic cookie corrispondono, l'utente ottiene il diritto di accesso; in caso contrario l'accesso viene negato.

SUN-DES-1

Il protocollo di autorizzazione SUN-DES-1, sviluppato da Sun Microsystems, è basato su Secure RPC (Remote Procedure Call) e richiede un supporto DES (Data Encryption Software). L'informazione chiave per l'autorizzazione è il nome di rete dell'utente. Questo nome viene codificato e inviato al server come parte del pacchetto di connessione. Il server decodifica l'informazione e quindi, se riconosce il nome di rete, abilita la connessione.

Questo protocollo offre un livello di protezione superiore a quello del MIT-MAGIC-COOKIE-1. Il nome di rete del proprio sistema non può essere in alcun modo utilizzato da altri utenti per accedere al server, mentre è possibile che un altro utente utilizzi il proprio magic cookie.

Il paragrafo “Abilitazione dell’accesso con SUN-DES-1”, in questo capitolo, spiega come autorizzare un altro utente ad accedere al proprio server aggiungendo il suo nome di rete alla lista degli accessi abilitati.

Modifica del protocollo di autorizzazione di default

Il protocollo di autorizzazione di default, MIT-MAGIC-COOKIE-1, può essere modificato per utilizzare SUN_DES-1, l’altro protocollo supportato, o per non utilizzare alcun meccanismo di controllo riferito all’utente. L’impostazione di default può essere modificata utilizzando le opzioni appropriate insieme al comando openwin. Ad esempio, per cambiare l’impostazione di default da MIT-MAGIC-COOKIE-1 a SUN-DES-1, avviare OpenWindows come segue:

```
$ openwin -auth sun-des
```

Per avviare OpenWindows senza il meccanismo di accesso riferito all’utente, digitare la riga di comando con l’opzione -noauth:

```
$ openwin -noauth
```

Attenzione – L’uso dell’opzione -noauth riduce la sicurezza del sistema. OpenWindows viene eseguito con il solo meccanismo di controllo riferito all’host, in quanto il server disattiva il meccanismo riferito all’utente. In questo modo, chiunque abbia accesso alle applicazioni di un dato sistema locale avrà accesso anche al relativo server.

Gestione degli accessi al server

Se OpenWindows non viene avviato con l’opzione -noauth (vedere “Modifica del protocollo di autorizzazione di default”), sono attivi sia il meccanismo di controllo riferito all’utente sia quello riferito all’host. Il server utilizza per primo il meccanismo riferito all’utente, e successivamente quello riferito

all'host. Il protocollo riferito all'utente impostato nella configurazione di default è MIT-MAGIC-COOKIE-1, mentre al meccanismo riferito all'host è associata una lista vuota. Ciò significa che è effettivamente attivo solo il meccanismo orientato all'utente. L'opzione `-noauth` ordina al server di disabilitare il meccanismo di controllo orientato all'utente e inizializza la lista degli host aggiungendovi l'host locale.

Per cambiare il meccanismo di controllo del server sono disponibili tre programmi: `xhost` e `xauth`. Questi programmi accedono a due file binari creati dal protocollo di autorizzazione, i quali contengono dati relativi ad ogni sessione. Uno dei file è riservato al server per uso interno, mentre l'altro si trova nella directory `$HOME` dell'utente:

- `.Xauthority` File autorizzazioni client

Il programma `xhost` permette di modificare la lista degli host con diritto di accesso al server. La lista può essere modificata con l'aggiunta o la cancellazione di altri host. Avviando il sistema con la configurazione di default — con la lista di accesso degli host vuota — e utilizzando `xhost` per aggiungere il nome di un sistema, di fatto si riduce il livello di sicurezza del sistema. Il server consentirà infatti l'accesso sia all'host inserito che a tutti gli utenti che specifichino il protocollo di autorizzazione di default. Per maggiori informazioni sul livello di protezione del meccanismo riferito all'host, vedere "Accesso riferito all'host".

Il programma `xauth` accede ai dati del protocollo di autorizzazione contenuti nel file `.Xauthority` del client. Per consentire ad altri utenti di accedere al proprio server o al server al quale si è connessi, è possibile estrarre queste informazioni dal proprio file `.Xauthority` in modo che tali utenti le iseriscano nel loro file `.Xauthority`.

Il paragrafo "Abilitazione dell'accesso con MIT-MAGIC-COOKIE-1" contiene alcuni esempi relativi all'uso di `xhost` e `xauth`.

File di autorizzazioni dei client

Il file che contiene le autorizzazioni dei client è `.Xauthority`. I dati sono contenuti nella forma seguente:

```
protcl_connessione   protcl_autorizzazione   dati_autorizzazione
```

Per default, `.Xauthority` contiene `MIT-MAGIC-COOKIE-1` come *protcl_autorizzazione* e istruzioni relative alla sola visualizzazione locale come *protcl_connessione* e *dati_autorizzazione*. Ad esempio, il file `.Xauthority` dell'host generico *host* potrebbe contenere le righe seguenti:

```
host:0          MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
hostlocale:0    MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
host/unix:0     MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
```

All'avvio dell'applicazione client, il sistema legge una riga corrispondente al *protcl_connessione* dal file `.Xauthority` e invia al server il *protcl_autorizzazione* e i *dati_autorizzazione* come parte del pacchetto di connessione. Nella configurazione di default, `xhost` restituisce liste di accesso vuote per gli host e dichiara abilitata l'autorizzazione di accesso.

Se il protocollo di autorizzazione di default è stato sostituito con `SUN-DES-1`, il file `.Xauthority` contiene `SUN-DES-1` come *protcl_autorizzazione* e il nome di rete dell'utente come *dati_autorizzazione*. Il nome di rete si presenta come segue:

```
unix.idutente@dominioNIS
```

Ad esempio, il file `.Xauthority` dell'host generico *host* può contenere le righe indicate nell'esempio seguente, dove unix.*15339@EBB.Eng.Sun.COM* è il nome di rete dell'utente indipendente dal sistema:

```
host:0          SUN-DES-1          "unix.15339@EBB.Eng.Sun.COM"
hostlocale:0    SUN-DES-1          "unix.15339@EBB.Eng.Sun.COM"
host/unix:0     SUN-DES-1          "unix.15339@EBB.Eng.Sun.COM"
```

Nota – Se non si conosce il proprio nome di rete o il nome di rete indipendente dal sistema, richiederlo all'amministratore di sistema.

Abilitazione dell'accesso con `MIT-MAGIC-COOKIE-1`

Se si utilizza il protocollo di autorizzazione MIT-MAGIC-COOKIE-1, per aggiungere un nuovo utente alla lista di accesso del server procedere come segue:

1. **Sul sistema che esegue l'applicazione server, eseguire `xauth` per estrarre in un file una voce corrispondente a *nomehost:0*.**
   In questo esempio, *nomehost* è *host* e il file è *xauth.info*:

```
$ $OPENWINHOME/bin/xauth nextract - host:0 > $HOME/xauth.info
```

2. **Inviare il file contenente questa riga all'utente che ha richiesto l'accesso (usando Posta Elettronica, `rcp` o un altro metodo di trasferimento file).**

Nota – L'invio del file contenente le informazioni di autorizzazione tramite posta elettronica è un metodo più sicuro rispetto all'uso di `rcp`. Se tuttavia si utilizza `rcp`, *non* collocare il file in una directory facilmente accessibile ad altri utenti.

---

3. **L'utente ricevente dovrà inserire la riga nel proprio file `.Xauthority`.**
   In questo esempio, l'utente designato inserisce `xauth.info` nel proprio file `.Xauthority`:

```
$ $OPENWINHOME/bin/xauth nmerge - < xauth.info
```

Nota – I *dati_autorizzazione* si riferiscono a una sessione specifica; ciò significa che sono validi solo per la sessione corrente del server.

Abilitazione dell'accesso con `SUN-DES-1`

Se si utilizza il protocollo di autorizzazione SUN-DES-1, per abilitare un nuovo utente ad accedere al server procedere come segue:

1. **Sul sistema che esegue il programma server, eseguire** `xhost` **per permettere al server di riconoscere il nuovo utente.**
   In questo esempio, il nuovo utente *nuovo_ut* viene abilitato ad accedere al sistema *host*:

```
$ xhost + nuovo_ut@
```

2. **Il nuovo utente deve eseguire** `xauth` **per aggiungere la nuova istruzione al proprio file** `.Xauthority`**.**
   In questo esempio, il nome di rete del nuovo utente indipendente dal sistema è unix.*15339@EBB.Eng.Sun.COM*. Si noti che questo comando deve essere scritto in un'unica riga, senza ritorni a capo. Dopo il simbolo di pipe, digitare uno spazio seguito dal resto del comando.

```
$ echo 'add host:0 SUN-DES-1 "unix.15339@EBB.Eng.Sun.COM"' |
$OPENWINHOME/bin/xauth
```

Nell'esempio, la lunghezza della riga genera un ritorno a capo; digitando il comando sul sistema, tuttavia, la riga dovrebbe comparire intera.

Esecuzione di applicazioni client in modo remoto, o esecuzione locale con un diverso nome utente

Le applicazioni client X utilizzano il valore della variabile d'ambiente `DISPLAY` per ricavare il nome del server al quale si devono collegare.

Per eseguire applicazioni client in modo remoto, o a livello locale con un diverso nome utente, procedere come segue:

1. **Sul sistema che esegue l'applicazione server, abilitare un nuovo utente.**
   A seconda del protocollo di autorizzazione utilizzato, seguire la procedura descritta in "Abilitazione dell'accesso con MIT-MAGIC-COOKIE-1" o in "Abilitazione dell'accesso con SUN-DES-1".

2. **Impostare la variabile d'ambiente** `DISPLAY` **sul nome dell'host che esegue il server.**
   In questo esempio, l'host è *hostremoto*:

```
$ DISPLAY=hostremoto:0
```

3. **Eseguire il programma client come indicato nell'esempio.**

```
$ programma_client&
```

Il client verrà visualizzato sul sistema remoto *hostremoto*.

Internetworking SPARC- DECnet (DNI)

Questa appendice descrive come collegare tra loro due reti che utilizzano gli ambienti OpenWindows e DECwindows™ attraverso il protocollo di trasporto NSP DECnet.

SPARC – Si noti che questo capitolo riguarda solo i sistemi SPARC. L'internetworking con DECnet è disponibile solo con DNI 8.x.

Per utilizzare DNI (DECnet Internetworking) si può procedere in due modi:

- Avviare un'applicazione client X11 su un sistema VAX (sotto il sistema operativo VMS®) e visualizzarla su un sistema OpenWindows
- Avviare un'applicazione client X11 su un sistema OpenWindows e visualizzarla su un sistema VAX

Questo capitolo descrive entrambe le possibilità, precedute da un'introduzione sui requisiti di configurazione DNI per i due sistemi.

Configurazione dell'internetworking DECnet

Per predisporre la comunicazione n internetworking con DECnet, procedere come segue:

1. **Abilitare una connessione tramite DNI.**
   Le librerie dei server e dei client OpenWindows utilizzano una versione dinamica della libreria di trasporto DNI `libdni`. Per fare in modo che le librerie dei server e dei client carichino `libdni`, occorre impostare la variabile d'ambiente `DNI_X_ENABLE` sulla directory in cui è installata la libreria `libdni.so`.

Nell'esempio seguente si suppone che DNI sia stato caricato con `pkgadd` nella posizione di default:

```
$ DNI_X_ENABLE=/opt/SUNWconn/dni/lib
```

2. **Avviare il server di OpenWindows.**
   Per default, il server di OpenWindows supporta il sistema di protezione "MIT-MAGIC-COOKIE". Questo meccanismo è riferito all'utente anziché all'host: in altre parole, si può stabilire quali utenti (anziché quali sistemi) si possono connettere al server. Nel modo di default, al comando `xhost` il sistema risponde con un elenco vuoto, e segnala soltanto che la protezione è attivata. Per disabilitare la protezione (e tornare al sistema di protezione delle precedenti versioni del server di OpenWindows) usare il comando `openwin` con l'opzione `-noauth`.

```
$ openwin -noauth
```

3. **Chiedere al proprietario del sistema su cui viene eseguito OpenWindows di eseguire il comando `xhost` per autorizzare il sistema DEC® VAX® a stabilire una connessione X11 con il server di OpenWindows.**
   Perché i client X11 si possano collegare al server di OpenWindows tramite DNI, gli indirizzi dei nodi DECnet devono essere mappati con i rispettivi nomi DECnet. Questo si ottiene creando ed inizializzando il database NCP. La stessa operazione deve essere eseguita anche sul sistema DEC VAX.

```
$ xhost decvax::
```

I due punti doppi specificano il trasporto DECNet.

Visualizzazione di un’applicazione client remota su un sistema OpenWindows

Per eseguire un’applicazione client X11 da VMS, effettuare il login nel sistema VAX usando il comando `dnilogin` di SunLink DNI. Per prima cosa, impostare la variabile d’ambiente `DISPLAY` in modo che il sistema locale operi come server X11 per il sistema remoto. Quindi avviare un’applicazione client X11 inserendo il nome del programma, rappresentato in questo esempio da *client_x11*. Per maggiori informazioni sull’uso di VMS, vedere il manuale *VMS DECwindows User’s Guide, Running Applications Across the Network.*

Ad esempio:

```
$ dnilogin decvax
    .
    .
    .
$ define DECW$DISPLAY OW_machine::0
$ spawn/nowait run client_x11
```

Visualizzazione di un’applicazione client remota su un sistema VAX

Per eseguire un’applicazione client X11 su un sistema OpenWindows e visualizzarla su un server DECwindows, impostare la variabile DISPLAY sul sistema VAX remoto.

Prima di avviare l’applicazione client X11 occorre compilare ed installare i font di OpenWindows sul server DECwindows. Questi font sono disponibili nella versione MIT X11R4 o nel pacchetto di font opzionali fornito insieme a OpenWindows versione 3.3. Per installare i font appropriati sul server DECwindows, procedere come segue:

1. **Installare i sorgenti dei font opzionali di OpenWindows (o i sorgenti dei font della versione MIT X11R4) sul sistema OpenWindows.**

2. **Seguire le istruzioni per l’installazione dei font riportate nel manuale** *OpenWindows Server Programmer’s Guide*.

3. **Copiare i sorgenti dei font in una directory del sistema VAX.**

```
$ cd $OPENWINHOME/share/src/fonts/misc
$ dnicp *.bdf 'decvax::[dirvax]'
```

4. **Compilare i font per il cursore sul sistema VAX.**
   Verrà creata una serie di file: `olcursor.decw$font;1` `olglyph10.decw$font;1`...

```
$ font olcursor.bdf
$ font olglyph10.bdf
...
```

5. **Copiare i font nella directory** `sysfont`:

```
$ set def sys$sysroot:[sysfont.decw.user_cursor16]
$ copy [dirvax]olcursor.decw$font;1 *
```

Nota – Per copiare i font nella directory `sysfont`, occorre accedere al sistema DECVAX come utente "system".

---

6. **Ripetere le operazioni 2-4 anche per gli altri font del cursore e per i font Lucida collocati in** `$OPENWINHOME/share/src/fonts/75dpi` **e** `$OPENWINHOME/share/src/fonts/100dpi`**.**

Nota – I font Lucida devono essere installati in `sys$sysroot:[sysfont.decw.user_75dpi]` e `sys$sysroot:[sysfont.decw.user_100dpi]`.

L'elenco seguente contiene il set minimo dei font richiesti per eseguire le applicazioni DeskSet di OpenWindows. Se per le applicazioni si utilizzano i font di default, non sarà necessario installarne altri. È tuttavia possibile installare un numero di font superiore a quello richiesto.

- olcursor.bdf
- olglyph10.bdf
- olglyph12.bdf

- olglyph14.bdf
- olglyph19.bdf
- luBS08.bdf
- luBS10.bdf
- luBS12.bdf
- luBS14.bdf
- luRS08.bdf
- luRS10.bdf
- luFS12.bdf
- lutBs12.bdf
- lutRS10.bdf
- lutRS12.bdf

7. Riavviare il server di DECwindows.

8. Per verificare che i font siano stati installati, richiedere l'elenco dei font disponibili nel server di DECwindows:

```
$ DISPLAY=decvax::0
$ xlsfonts | grep Sun     (font cursore)
$ xlsfonts | grep Lucida
```

9. Verificare di avere abilitato il nodo OpenWindows per la visualizzazione sul server di DECwindows usando il menu Security del DECwindows Session Manager.

10. Avviare un'applicazione X11 (ad esempio, un'applicazione DeskSet di OpenWindows).

```
$ DISPLAY=decvax::0
$ mailtool
```

Nota – `DNI_X_ENABLE` deve essere impostato sulla directory della libreria di trasporto DNI `libdni`. Vedere a questo riguardo il punto 1 del paragrafo "Configurazione dell'internetworking DECnet" in questo capitolo.

Se si riceve un messaggio di errore come quello illustrato nell'esempio seguente, per avviare l'applicazione è necessario installare il font specificato nel server di DECwindows.

```
XView warning: Cannot load font '-b&h-lucida-medium-r-*-*-
*-80-*-*-*-*-*-*' (Font package)
```

Questo messaggio di errore segnala che occorre installare il font `luRS10.bdf`.

Per maggiori informazioni sui font, vedere il manuale *OpenWindows Server Programmer's Guide*.

Installazione di modem e stampanti

Installazione di un modem

Un *modem* è un dispositivo che permette a un sistema di trasmettere e ricevere informazioni attraverso una linea telefonica. Per installare un modem, l'applicazione grafica Administration Tool offre un metodo estremamente semplice. Procedere come segue:

1. **Collegare il modem.**
   In genere, il collegamento di un modem richiede le seguenti operazioni:

   - Collegare fisicamente il modem al sistema o, nel caso dei modem su scheda, installare la scheda.
   - Impostare gli switch necessari e/o configurare la velocità (baud rate), la porta di comunicazione e gli altri parametri richiesti.
   - Se necessario, collegare il modem o il relativo adattatore a una presa di rete.

   Per ulteriori informazioni sulle operazioni di collegamento, consultare la documentazione del modem e il manuale di installazione del sistema.

2. Diventare utente root.

L'uso di Administration Tool è riservato ai membri del gruppo speciale di amministratori di sistema UNIX (GID 14) e all'utente *root*. Quest'ultimo è un utente dotato di autorizzazioni speciali per la modifica dei file di sistema.

Per diventare utente root, usare il comando `su`:

```
$ su
Password:       (inserire la password di root)
#
```

Se il profilo dell'utente root è protetto da una password, sarà necessario inserirla alla richiesta del sistema. In caso contrario sarà sufficiente premere Return. Per conoscere la password di root, rivolgersi all'amministratore di sistema.

3. Avviare Administration Tool.

Digitare:

```
# /usr/bin/admintool &
#
```

Dopo alcuni secondi comparirà la finestra di Administration Tool.
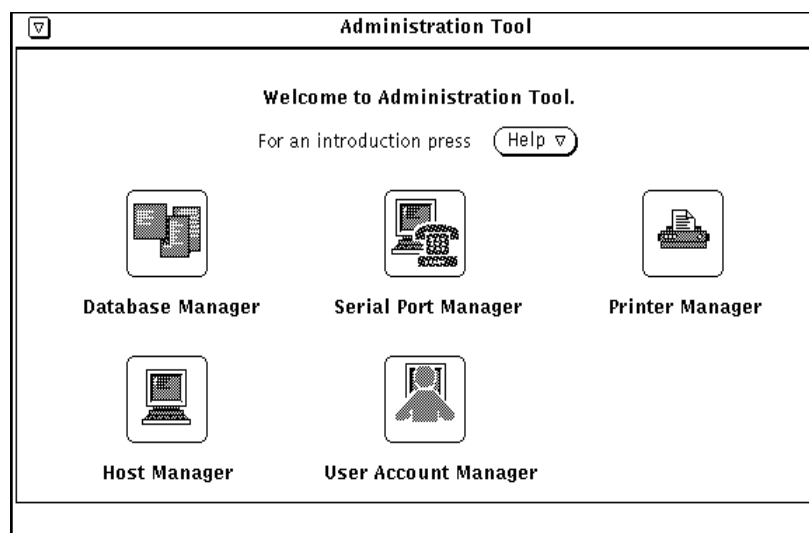
Figura E-1 L'utility Administration Tool

4. **Selezionare il gestore delle porte seriali (Serial Port Manager).**
   Per avviare Serial Port Manager, fare clic sull'icona corrispondente nella finestra di Administration Tool. Viene aperta una nuova finestra con i comandi e le opzioni per la gestione dei modem e dei terminali.

5. **Seguire le istruzioni della guida online.**
   La guida online di Serial Port Manager contiene istruzioni dettagliate sull'uso dell'utility per l'installazione di un modem. Per visualizzare la guida online, scegliere Help… dalla finestra di Serial Port Manager.

Per ulteriori informazioni sui modem

Per ulteriori informazioni sull'installazione e la configurazione dei modem, vedere il manuale *Peripherals Administration*, incluso in *Solaris 2.4 System Administrator AnswerBook*.

Installazione di una stampante

Il metodo più semplice per installare una stampante è quello offerto dall'applicazione grafica Administration Tool. Le istruzioni seguenti spiegano come installare una stampante per l'utilizzo sul sistema locale. Per informazioni e istruzioni sulla configurazione delle stampanti in rete, vedere il manuale *Peripherals Administration*, incluso in *Solaris 2.4 System Administrator AnswerBook.*

Per installare una stampante locale, procedere come segue:

1. **Collegare la stampante.**
   In genere, il collegamento di una stampante richiede le seguenti operazioni:

   - Collegare fisicamente la stampante al sistema
   - Impostare gli switch necessari e/o configurare la velocità (baud rate), la porta di comunicazione e gli altri parametri richiesti.
   - Collegare la stampante a una presa di rete.

   La stampante può essere collegata e accesa prima o dopo la procedura di configurazione con Administration Tool. Per informazioni sull'impostazione degli switch e sui cavi richiesti, vedere il manuale della stampante e il manuale di installazione del sistema. Il cavo della stampante viene di norma collegato a una porta seriale, ma in alcuni casi può essere richiesto il collegamento a una porta parallela.

2. **Diventare utente root.**
   L'uso di Administration Tool è riservato ai membri del gruppo speciale di amministratori di sistema UNIX (GID 14) e all'utente *root*. Quest'ultimo è un utente dotato di autorizzazioni speciali per la modifica dei file di sistema.

   Per diventare utente root, usare il comando `su`:

```
$ su
Password:       (inserire la password di root)
#
```

   Se il profilo dell'utente root è protetto da una password, sarà necessario inserirla alla richiesta del sistema. In caso contrario sarà sufficiente premere Return. Per conoscere la password di root, rivolgersi all'amministratore di sistema.

3. **Avviare Administration Tool.**
   Digitare:

```
# /usr/bin/admintool &
#
```

   Dopo alcuni secondi compare la finestra di Administration Tool. (Vedere la Figura E-1 a pagina 195.)

4. **Selezionare il gestore delle stampanti (Printer Manager).**
   Per avviare Printer Manager, fare clic sull'icona corrispondente nella finestra di Administration Tool. Verrà aperta una nuova finestra con i comandi e le opzioni per la gestione delle stampanti.

5. **Seguire le istruzioni della guida online.**
   La guida online di Printer Manager contiene istruzioni dettagliate sull'uso dell'utility per l'installazione di una stampante. Per visualizzare la guida online, scegliere Help... dalla finestra di Printer Manager.

Per ulteriori informazioni sulle stampanti

Per informazioni dettagliate sull'installazione e la gestione delle stampanti, vedere il manuale *Peripherals Administration*, incluso in *Solaris 2.4 System Administrator AnswerBook.*

Indice analitico

D

P

W

X

Y